Sparsity-Based Algorithms for Line Spectral Estimation

Hansen, Thomas Lundgaard

DOI (link to publication from Publisher):
10.5278/vbn.phd.tech.00037

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

SPARSITY-BASED ALGORITHMS FOR LINE SPECTRAL ESTIMATION

BY
THOMAS LUNDGAARD HANSEN

DISSERTATION SUBMITTED 2018

AALBORG UNIVERSITY
DENMARK

Sparsity-Based Algorithms for Line Spectral Estimation

Ph.D. Dissertation

Thomas Lundgaard Hansen

Section of Wireless Communication Networks
Department of Electronic Systems
Technical Faculty of IT and Design
Aalborg University, Denmark

Dissertation submitted: March 2, 2018

PhD supervisor: Prof. Bernard H. Fleury
Aalborg University

Assistant PhD supervisor: Prof. Bhaskar D. Rao
University of California at San Diego

PhD committee: Professor Søren Holdt Jensen (chairman)
Aalborg University

Professor Mario Figueiredo
University of Lisbon

Professor Dirk Slock
Eurecom

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-170-5

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Typeset by the author using LaTeX.

Printed in Denmark by Rosendahls, 2018

Abstract

Line spectral estimation is a classical problem in signal processing. It has found broad application in for example array processing, wireless communication, localization, radar, radio astronomy and audio. In the last decade we have seen significant research into sparsity-based processing techniques. The use of sparsity-based techniques has allowed for advances to both the design and analysis of algorithms for line spectral estimation. In this thesis we study the design of such algorithms.

The uniting theme of our contributions is the design of algorithms that make sparsity-based line spectral estimation viable in practice. First, it is demonstrated that these schemes can be applied to the estimation of wireless channels of not only specular but also of diffuse nature. We attribute that to a low-rank property of the channel covariance matrix, a concept that we elaborate on.

The design of algorithms for sparsity-based line spectral estimation in a general context is then considered. The obtained algorithms are computationally feasible for much larger problems than what concurrent algorithms can practically deal with and show high estimation accuracy.

Resumé

Linje spektral estimering er et klassisk problem indenfor signal behandling. Det har fundet bred anvendelse indenfor for eksempel databehandling i lyd- og antennegruppe systemer, trådløs kommunikation, lokalisering, radar og radio astronomi. I det seneste årti har vi set en stor forskningsinteresse i teknikker baseret på sparsommelighed. Ved brug af disse teknikker har man opnået resultater vedrørende både designet og analysen af algoritmer til linje spektral estimering. I denne afhandling undersøger vi designet af sådanne algoritmer.

Det sammenfattende tema i afhandlingen er designet af algoritmer der gør anvendelsen af sparsommeligheds-teknikker realistisk i praksis. Det demonstreres først at disse metoder kan anvendes til estimering af trådløse kanaler der ikke kun består af separate komponenter, som intuitionen foreslår, men også kanaler med diffuse komponenter. Dette fænomen tilskrives en lav-rang egenskab af kanalens kovarians matrix. Der fremvises en uddybende analyse af dette koncept.

Derefter betragtes designet af algoritmer til linje spektral estimering baseret på sparsommeligheds-teknikker i en generel kontekst. Der fremvises algoritmer der beregningsmæssigt er praktiske at anvende på langt større problemer end nuværende algoritmer tillader. De fremviste algoritmer opnår samtidigt stor nøjagtighed af estimaterne.

Contents

Thesis Details

Thesis Title:	Sparsity-Based Algorithms for Line Spectral Estimation
Ph.D. Student:	Thomas Lundgaard Hansen
Supervisors:	Prof. Bernard H. Fleury, Aalborg University
	Prof. Bhaskar D. Rao, University of California at San Diego

The main body of this thesis consists of the following papers.

[A] T. L. Hansen, P. B. Jørgensen, M.-A. Badiu and B. H. Fleury, "A turbo receiver for OFDM with sparsity-based parametric channel estimation," submitted to *IEEE Transactions on Signal Processing*.

[B] T. L. Hansen, B. H. Fleury and B. D. Rao, "Superfast line spectral estimation," accepted for publication in *IEEE Transactions on Signal Processing*.

[C] T. L. Hansen and T. L. Jensen, "A fast interior point method for atomic norm soft thresholding," submitted to *Applied and Computational Harmonic Analysis*.

In addition to the main papers, the following publications have been co-authored by the doctoral candidate in connection with the study.

[1] T. L. Hansen, M.-A. Badiu, B. H. Fleury and B. D. Rao, "A sparse Bayesian learning algorithm with dictionary parameter estimation," in *Proceedings of the 8th IEEE Sensor Array and Multichannel Signal Processing Workshop,* pp. 385–388, Jun. 2014.

[2] M.-A. Badiu, T. L. Hansen and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Transactions on Signal Processing*, vol. 65, no. 9, pp. 2247–2261, May 2017.

[3] D. A. Wassie, T. L. Hansen, I. L. Rodriguez, G. Berardinelli, F. M. L. Tavares, T. B. Sørensen and P. Mogensen, "An agile multi-node multi-antenna wireless channel sounding system," to be submitted to *IEEE Access*.

Preface

The dissertation You are now reading is the culmination of approximately four years worth of hard work. The underpinning for the work was laid already during my time as a master student at Aalborg University where I cooperated with both of my supervisors and other key people. It was during this time that I was introduced to the subjects of sparse modelling and its use for line spectral estimation. Towards the end of my time as a master student we applied for funding to further study the subject. The Danish Council for Independent Research responded positively and granted money that have supported me throughout my time as a doctoral student.

The material in this dissertation expands on our understanding of the application of sparse estimation to line spectra; in particular with respect to the underlying mechanisms at play in the design of such estimation schemes. In the dissertation I have sought to convey this intuition and understanding of the methods without skipping on mathematical rigour.

The thesis is presented in the form of a collection of papers. This means that the main body of the thesis is a number of papers that have been written during the PhD study. The contents of the papers are reproduced verbatim, with only the layout modified to match that of the thesis. For this reason the reader should not expect consistency of notation between the individual papers.

I hope you will find the dissertation to be an interesting read.

Thomas Lundgaard Hansen
Aalborg University, March 2, 2018

Acknowledgements

I would like to take this chance to express my gratitude towards my supervisor Prof. Bernard H. Fleury. You always gave me the freedom and encouragement to pursue the research avenues that I found the most interesting. I also have gratitude towards Prof. Bhaskar D. Rao, who hosted me at the University of California San Diego on two separate occasions. Your broad knowledge and ingenious mindset is inspiring.

Also my colleagues at Aalborg University deserve a big "thank you" for their influence during my time as a doctoral student. There are a few people I would like to mention by name: Carles, for always being a source of sound advice on all aspects of life as a Ph.D. student, be they of technical nature or not. Troels, for helping me understand the importance of thinking about our methodologies in a broad context. Mihai, for patiently helping me understand the details of variational inference and what research is in our field. Tobias, for a very interesting collaboration towards the end of the study that introduced me to the design of convex optimization algorithms in general and interior point methods in particular.

Finally, I would like to express my unconditional gratitude to the persons who have accepted to be members of the Ph.D. assessment committee: Prof. Søren H. Jensen, Prof. Mário A. T. Figueiredo and Prof. Dirk Slock.

Part I

Introductory Chapters

Chapter 1

Introduction

One of the core interests in signal processing and machine learning is *signal analysis*. In science and technology we seek to understand the world around us by observing it. When the observations, or measurements, are taken in a systematic way we refer to the outcome as a signal. The purpose of signal analysis is to extract information and understanding about the world from such signals.

A typical approach to signal analysis is via *parametric modelling*. A model is a simplified mathematical description of how the world approximately behaves. A parametric model is one which can be completely characterized by a finite number of parameters. It is often, but not always, the case that the parameters of such models have a direct meaning or interpretation in the physical world. In these cases we can gain understanding about the physical world by estimating the parameters of the parametric model. In this work we focus on a special instance of signal analysis, namely parametric estimation of *line spectra*.

1.1 Spike Recovery

Before going into the details of line spectral estimation, we take a small detour to illustrate some key concepts. On an abstract level the estimation of line spectra deals with the recovery of fine-scale details from only coarse-scale and noisy observations of an object. In order to illustrate this task, we consider a simple illustrative example of spike recovery in image processing. In this analogy, the task is to locate spikes in an image from noisy and blurred observations. In practical applications it is almost never known a-priori how many spikes there are and so this number must also be estimated. The spikes here illustrate the fine-scale details which must be recovered. Such a problem for example exists in astronomy, where the spikes are distant stars to be

identified.

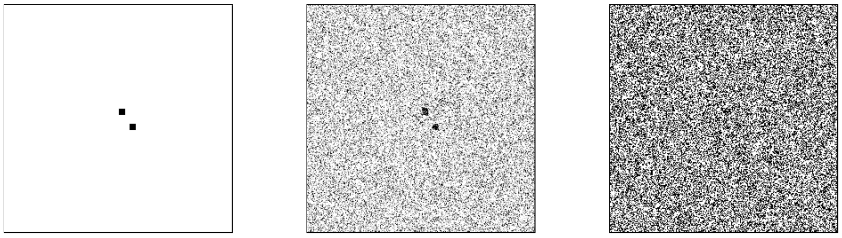

Fig. 1.1: Spike recovery from noisy observations in image processing, with noise-free (left), medium noise level (center) and high noise level (right).

Consider first the illustration of the recovery of spikes (black points) from noisy observations in Fig. 1.1. The noise level in the noise-free and medium noise level cases makes it easy to determine the number of spikes and their location. In the case with high noise level, it is hard to determine with the naked eye how many spikes there are; and even if it is known that there are two, their location cannot easily be determined.

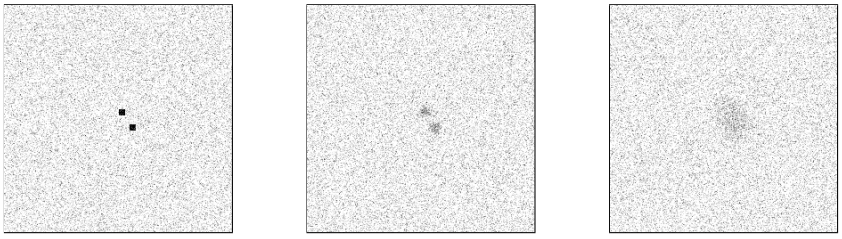

Fig. 1.2: Spike recovery from noisy and coarse-scale (blurred) observations in image processing, with no blurring (left), medium blurring (center) and significant blurring (right).

As mentioned, only coarse-scale observations are available, i.e., the fine details of the object under consideration are distorted by the observation process. In the image processing analogy this effect could be caused by optical distortions (out of focus, diffraction limit, etc.) or motion blur due to relative motion between observed and observing entities [1]. These effects are modelled as a blur filter as demonstrated in Fig. 1.2. It is clear that the amount of blurring determines the ability to resolve closely located spikes.

Intuitively we solve the above tasks by identifying areas of the picture that are relatively dark. Then, when is an area sufficiently dark to assert that there is indeed one – or more! – spikes? This question essentially amounts to that of model order selection and in this thesis we use sparse regularization to (algorithmically) answer it.

1.2 Line Spectral Estimation

We are now ready to introduce the problem of line spectral estimation (LSE). An informal definition of *spectral estimation* is given by Stoica and Moses in their seminal book on spectral analysis of signals [2]: "From a finite record of a stationary data sequence, estimate how the total power is distributed over frequency." With this definition in mind consider a signal $x(t)$ whose spectrum (Fourier transform) $X(f)$ is a sum of spikes,

$$X(f) = \sum_{l=0}^{L-1} \alpha_l \delta(f - f_l), \tag{1.1}$$

where $\delta(\cdot)$ is the Dirac delta function. Spectra of the form (1.1) are known as line spectra. The lth spike is located at frequency f_l and has amplitude given by the coefficient α_l. In the image processing example the locations of the spikes are analogous to the frequencies $\{f_l\}$.

Since $X(f)$ is the spectrum of $x(t)$ [3, 4], LSE must be about estimating the parameters of the model (1.1). To be specific, the number of spikes (sinusoids) L, the frequencies $\{f_l\}$ and the coefficients $\{\alpha_l\}$ must be estimated. However, the definition of spectral estimation given by Stoica and Moses rather suggests that the objective of LSE is to estimate L, the frequencies $\{f_l\}$ and the *powers* $\mathbb{E}\left[|\alpha_l|^2\right]$ of the coefficients $\{\alpha_l\}$. In this work we make no distinction between these two definitions of LSE. The challenging task in LSE is to estimate L and the frequencies $\{f_l\}$. Once these are obtained the coefficients $\{\alpha_l\}$ can easily be estimated with a least-squared approach. If the powers of $\{\alpha_l\}$ are of interest they can for example be estimated by taking the magnitude-square of the entries in $\{\alpha_l\}$.

The observation process through which we can observe the object $X(f)$ is obtained by taking a limited number of noisy time-domain samples, i.e., we observe a vector $\boldsymbol{y}$ with entries

$$y_n = x(n\Delta_T) + w_n, \qquad n = 0, \ldots, N-1, \tag{1.2}$$

where Δ_T is a sampling time and w_n is a noise process. The vector $\boldsymbol{y}$ is the "finite record" mentioned by Stoica and Moses. This observation process distorts the fine details of the line spectrum because only a limited number of time-domain samples are observed. Indeed, by discrete-time Fourier transforming the observation vector $\boldsymbol{y}$ we get that the frequency-domain representation (1.1) has been blurred (specifically it has been circularly convolved with a sinc-filter). Fig. 1.3 gives an example of a line spectrum (blue spikes) and the corresponding frequency-domain representation of the observed samples (red line). It is clear that the number of spikes and their location cannot be immediately identified from the observation even though there is no noise in this example. In this thesis we are concerned with algorithms that can

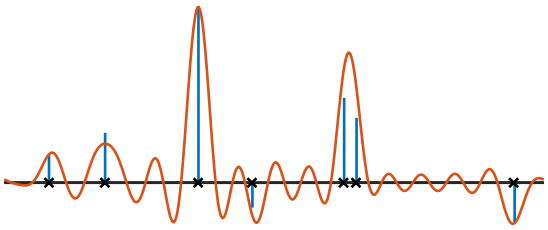

Fig. 1.3: Recovery of line spectra (blue spikes) from only coarse-scale observations (red line). Frequencies recovered by the superfast LSE algorithm (Paper B) are marked (black crosses).

solve this task. As an example we have run the superfast LSE algorithm (Paper B) on the observation in Fig. 1.3. The algorithm correctly recovers both the number of spikes and their location (black crosses).

In the image processing example, we concluded that the noise mainly affects the detection of spikes, independently of their separation. On the other hand, the degree to which the observation process distorts the fine details dictates the resolution with which closely located spikes can be separated. This intuition can to a large degree be transferred directly to the LSE problem and in our experience goes a very long way to explain the estimation accuracy which can be achieved.

Now consider a scenario where a limited number of noisy samples of the transfer function (i.e., frequency-domain) of a linear time-invariant filter are observed. Also assume that the corresponding impulse response (i.e., time-domain) can be modelled as a sum of spikes analogously to (1.1). Let's assume that we are interested in estimating the impulse response from the observed samples of the transfer function. Due to the time-frequency duality of the Fourier transform such a problem is equivalent to that of LSE (ignoring a few insignificant details).

There exists a multitude of estimation problems that, as above, can be cast in a similar form as LSE and for this reason it is ubiquitous in signal processing literature. Example applications include direction of arrival estimation using sensor arrays [5, 6], bearing and range estimation in synthetic aperture radar [7], estimation of Doppler shifts [8], carrier offset estimation in communication systems [9] and simulation of atomic systems in molecular dynamics [10]. The work in this thesis is motivated by an application for estimation of wireless communication channels [11–13]; we introduce the details of that application in Chapter 3. The algorithms we develop in the thesis are generally applicable to both wireless channel estimation and other instances of LSE.

1.3 Focus of The Thesis

For the last decade, sparse processing has been a hot topic in the field of signal processing. This interest in sparsity is in large parts due to its central use in compressed sensing [14–16]. The application of sparsity-based methods to LSE has also gained attraction. Sparsity-based approaches distinguish themselves by inherently estimating the number of sinusoids (the model order) while classical methods require a separate methods to estimate this number, for example using information-theoretic model order selection criteria [17–23]. There are also other cases where the sparsity-based methods have proven superior, for example when the coefficients $\{\alpha_l\}$ are statistically correlated [24].

The focus of the thesis is that of sparsity-based methods for LSE. Special attention is paid to the computational and algorithmic aspects of this class of methods. We first consider how this type of estimator can be used for estimation of wireless channels and incorporate the estimator into an iterative receiver design. The focus is then shifted away from that particular application and we develop general-purpose algorithms for LSE. These algorithms distinguish themselves from the state of the art by having significantly lower computational complexity while showing very high estimation accuracy.

1.4 Thesis Outline

The organization of the thesis is as follows.

In Part I we provide some introductory and background material to set the stage of the work: Chapter 2 presents some classical methods for LSE along with the motivation for continued research on this problem. The sparsity-based approach to LSE is introduced and it is shown that methods in this class can be understood in a unified framework of rank minimization. In Chapter 3 we elaborate on the application to wireless channel estimation. It is demonstrated that the rank minimization viewpoint gives some novel insights into the applicability of parametric channel estimators. In particular a rigorous argument is given for the applicability to channels that are not purely specular. Chapter 4 gives an overview of the content and the scientific contribution of each of the papers. We also look ahead and consider the future use of the developed methods along with a review of promising directions for future research.

Part II contains the research papers that have been produced during the PhD study.

Chapter 2

Context of the Work

2.1 Preliminaries

To begin our endeavours rewrite the signal model (1.2) into the form

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}, \qquad \boldsymbol{x} = \mathbf{A}(\boldsymbol{f})\boldsymbol{\alpha}, \tag{2.1}$$

where $\boldsymbol{x}$ is the signal vector and the frequencies and coefficients have been collected in the vectors $\boldsymbol{f} = (f_0, \ldots, f_{L-1})^{\mathrm{T}} \in [0,1)^L$ and $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{L-1})^{\mathrm{T}} \in \mathbb{C}^L$. The vector $\boldsymbol{w}$ is a noise vector. Throughout the thesis it is assumed that the entries of $\boldsymbol{w} \in \mathbb{C}^N$ are independent and identically distributed (i.i.d.) with each entry having a zero-mean complex Gaussian distribution[1] of variance σ^2. Where applicable, the variance of α_l is denoted γ_l and these variances are collected in the vector $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_{L-1})^{\mathrm{T}} \in (0,\infty)^L$. We define the vector $\boldsymbol{a}(f) \in \mathbb{C}^N$ which has nth entry $(\boldsymbol{a}(f))_n = \exp(-j2\pi nf)$ for $n = 0, \ldots, N-1$, i.e., it contains samples of a sinusoid with frequency f. The matrix $\mathbf{A}(\boldsymbol{f}) \in \mathbb{C}^{N\times L}$ is composed with the sinusoidal vectors as its columns, i.e., $\mathbf{A}(\boldsymbol{f}) = \big(\boldsymbol{a}(f_0), \ldots, \boldsymbol{a}(f_{L-1})\big)$.

The LSE problem is that of recovering from $\boldsymbol{y}$ the number of sinusoids L, the frequencies $\boldsymbol{f}$ and either the coefficients $\boldsymbol{\alpha}$ or their variances $\boldsymbol{\gamma}$.

In some applications multiple (say M) observations of the form (2.1) are available. That is known as the multiple measurement vector (MMV) case [25–27]. Here the frequency vector $\boldsymbol{f}$ and the variances $\boldsymbol{\gamma}$ are common while the coefficient vector $\boldsymbol{\alpha}$ and the noise vector $\boldsymbol{w}$ may vary between each obtained observation. If all the observation vectors are collected into a matrix $\mathbf{Y} \in \mathbb{C}^{N\times M}$, the MMV model can be written as

$$\mathbf{Y} = \mathbf{X} + \mathbf{W}, \qquad \mathbf{X} = \mathbf{A}(\boldsymbol{f})\mathbf{B}, \tag{2.2}$$

[1] Throughout the thesis we assume circular symmetry of all complex Gaussian distributions.

where $\mathbf{B} \in \mathbb{C}^{L \times M}$ contains the coefficient vectors $\boldsymbol{\alpha}$ as columns and $\mathbf{W} \in \mathbb{C}^{N \times M}$ is a noise matrix.

A naïve, first approach to LSE could be to calculate the (magnitude-square) Fourier transform of the observed signal. This approach is known as the *periodogram* method. Simply detecting the peaks of the periodogram suffers from the same resolution issue as is visible in Fig. 1.3. It is commonly asserted that the periodogram cannot resolve frequencies which are located closer than about $1/N$ [28]. We are interested in methods that can go significantly below this limit. Such methods are commonly known as *super-resolution* methods [29].

2.2 Classical Methods for Line Spectral Estimation

Due to the many instances in which the estimation of line spectra has an application, the problem has been attacked from many angles. The earliest work goes all the way back to a 1795 paper by Prony [30]. As witnessed from the list of references accompanying this introduction, much of the groundwork for our current understanding was done in the 1980s and early 1990s. We here shortly review some classical methods for LSE. By classical methods we mean those methods that do not directly use the notion of sparse modelling.

It would be futile to try and give a complete overview of the literature on LSE. We instead discuss two main classes of methods (subspace and maximum likelihood methods) with the intent to give an idea of the types of methods that have been considered along with some of their limitations. Among other approaches we mention higher order Yule-Walker [31, 32], matrix pencil [33, 34], maximum entropy [35–37], weighted subspace fitting [38, 39] and filter-bank [40–42] methods. Stoica and Moses [2] dedicate a complete chapter that provides an excellent description of many of the techniques for LSE. We refer to [2, 43, 44] as a starting point for further reading.

2.2.1 Subspace Methods

As the name suggests subspace methods work by analysing the signal and noise subspaces which can be obtained from an eigenvalue decomposition of the signal covariance matrix. This class of methods include Pisarenko's method [45], MUltiple SIgnal Classification (MUSIC) [46], Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) [47, 48] and the min-norm method [49].

Assume that the coefficients in $\boldsymbol{\alpha}$ are zero-mean random variables and consider the covariance matrix of $\boldsymbol{y}$ in (2.1),

$$\boldsymbol{\Sigma}_y = \boldsymbol{\Sigma}_x + \sigma^2 \mathbf{I}, \qquad \boldsymbol{\Sigma}_x = \mathbf{A}(f) \boldsymbol{\Gamma} \mathbf{A}^{\mathrm{H}}(f), \tag{2.3}$$

where $\mathbf{\Gamma} = \text{diag}(\boldsymbol{\gamma})$ is a diagonal matrix with the entries of $\boldsymbol{\gamma}$ on the diagonal. The eigenvalues $\{\lambda_n\}$ of $\mathbf{\Sigma}_{\boldsymbol{y}}$ (sorted in descending order) satisfy

$$\begin{cases} \lambda_n > \sigma^2 & \text{for } n = 0, \dots, L-1, \\ \lambda_n = \sigma^2 & \text{for } n = L, \dots, N-1. \end{cases} \tag{2.4}$$

We refer to the first L eigenvalues as the "signal eigenvalues" and the remaining $N - L$ ones as the "noise eigenvalues". Now write the eigenvalue decomposition of $\mathbf{\Sigma}_{\boldsymbol{y}}$ as

$$\mathbf{\Sigma}_{\boldsymbol{y}} = \mathbf{U}_{\text{s}} \mathbf{\Lambda}_{\text{s}} \mathbf{U}_{\text{s}}^{\text{H}} + \mathbf{U}_{\text{n}} \mathbf{\Lambda}_{\text{n}} \mathbf{U}_{\text{n}}^{\text{H}}, \tag{2.5}$$

where $\mathbf{\Lambda}_{\text{s}} = \text{diag}(\lambda_0, \dots, \lambda_{L-1})$ and $\mathbf{\Lambda}_{\text{n}} = \text{diag}(\lambda_L, \dots, \lambda_{N-1})$ contain the signal and noise eigenvalues. The matrix $\mathbf{U}_{\text{s}} \in \mathbb{C}^{N \times L}$ has as columns the eigenvectors associated to the signal eigenvalues and $\mathbf{U}_{\text{n}} \in \mathbb{C}^{N \times N-L}$ has as columns the eigenvectors associated to the noise eigenvalues.

The key property, upon which all subspace methods are build, is that the "noise subspace" spanned by the columns of $\mathbf{U}_{\text{n}}$ is orthogonal to the columns of $\mathbf{A}(\boldsymbol{f})$. To see this we write

$$\mathbf{\Sigma}_{\boldsymbol{y}} \mathbf{U}_{\text{n}} = \left(\mathbf{U}_{\text{s}} \mathbf{\Lambda}_{\text{s}} \mathbf{U}_{\text{s}}^{\text{H}} + \mathbf{U}_{\text{n}} \mathbf{\Lambda}_{\text{n}} \mathbf{U}_{\text{n}}^{\text{H}} \right) \mathbf{U}_{\text{n}} = \sigma^2 \mathbf{U}_{\text{n}}. \tag{2.6}$$

But note that from (2.3) we also have

$$\mathbf{\Sigma}_{\boldsymbol{y}} \mathbf{U}_{\text{n}} = \mathbf{A}(\boldsymbol{f}) \mathbf{\Gamma} \mathbf{A}^{\text{H}}(\boldsymbol{f}) \mathbf{U}_{\text{n}} + \sigma^2 \mathbf{U}_{\text{n}} \tag{2.7}$$

and so $\mathbf{A}(\boldsymbol{f}) \mathbf{\Gamma} \mathbf{A}^{\text{H}}(\boldsymbol{f}) \mathbf{U}_{\text{n}} = \mathbf{0}$. Since $\mathbf{A}(\boldsymbol{f}) \mathbf{\Gamma}$ is full rank it follows that

$$\mathbf{A}^{\text{H}}(\boldsymbol{f}) \mathbf{U}_{\text{n}} = \mathbf{0}, \tag{2.8}$$

and so $\boldsymbol{a}^{\text{H}}(\tilde{f}) \mathbf{U}_{\text{n}} = \mathbf{0}$ if $\tilde{f}$ is an entry in $\boldsymbol{f}$. It can be shown that the condition $\boldsymbol{a}^{\text{H}}(\tilde{f}) \mathbf{U}_{\text{n}} = \mathbf{0}$ is not only necessary but also sufficient for $\tilde{f}$ to be an entry in $\boldsymbol{f}$ [2].

To give a flavour of how subspace methods exploit the above property to estimate the frequencies, we here describe how the MUSIC algorithm[2] proceeds:

1. Estimate the covariance matrix $\mathbf{\Sigma}_{\boldsymbol{y}}$ from the observed data.
2. Calculate the eigenvalue decomposition of the estimated covariance matrix and extract the estimated noise eigenvectors in $\hat{\mathbf{U}}_{\text{n}}$.

[2] The variant presented here is sometimes referred to as *spectral* MUSIC to contrast it with *root* MUSIC [2], which we do not discuss here.

3. Estimate the frequencies $\boldsymbol{f}$ as the L locations of the peaks of the MUSIC pseudo spectrum

$$P_{\text{music}}(f) = \frac{1}{\|\hat{\mathbf{U}}_{\text{n}}^{\text{H}} \boldsymbol{a}(f)\|^2}, \qquad f \in [0, 1). \tag{2.9}$$

The last step above identifies those frequencies for which the Euclidean distance between $\hat{\mathbf{U}}_{\text{n}}^{\text{H}} \boldsymbol{a}(f)$ and the zero vector is smallest. It is never exactly zero because an estimate of the noise eigenvectors are used.

The MUSIC algorithm and other subspace algorithms generally work very well and are widely used in practice. They are able to resolve sinusoidal components that have frequency separation much below $1/N$, i.e., the resolution limit of the periodogram. Numerical evidence also suggests that the accuracy of the estimates is close to the Cramér-Rao bound [2] which is the best possible performance of the class of unbiased estimators. There are, however, a few disadvantages to this class of methods:

- Their performance degrades if the statistical assumptions of white noise and no statistical correlation between the coefficients $\{\alpha_l\}$ are violated [2, 50].
- They assume that the number of sinusoids L is known. That is usually not the case in practical applications. This means that the subspace methods must be used in conjunction with a method to estimate the number of sinusoids, see [17–22] for a few examples of how that can be done.
- The subspace methods require as their input an estimate of the signal covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{y}}$. There are a number of ways to obtain such an estimate, see [43, 51–53]. In the MMV case the sample covariance matrix may suffice. If only a few or just a single observation vector is available the formation of the covariance matrix may require that the observation vector $\boldsymbol{y}$ is subdivided to artificially "create" multiple observation vectors of shorter length. In such cases the subspace methods suffer a degradation of estimation accuracy and resolution ability.

2.2.2 Maximum Likelihood Methods

Another class of classical approach to LSE is via maximum likelihood (ML) estimation. The idea is to estimate the parameters as those which are most likely to explain the observed data. There are two main formulations referred to as *deterministic* ML and *stochastic* ML [6]. To simplify the discussion we assume that the noise variance σ^2 is known in this section.

In deterministic ML [54] the coefficient vector $\boldsymbol{\alpha}$ is considered to be a deterministic parameter to be estimated. Under the white Gaussian noise

assumption the likelihood function $p(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{\alpha})$ is a multivariate complex Gaussian probability density function (pdf) with mean $\mathbf{A}(\boldsymbol{f})\boldsymbol{\alpha}$ and covariance matrix $\sigma^2\mathbf{I}$. The deterministic ML estimator then reads

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\alpha}}) = \arg\max_{\boldsymbol{f},\boldsymbol{\alpha}} \ln p(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{\alpha}) \tag{2.10}$$

$$= \arg\min_{\boldsymbol{f},\boldsymbol{\alpha}} \|\boldsymbol{y} - \mathbf{A}(\boldsymbol{f})\boldsymbol{\alpha}\|^2. \tag{2.11}$$

The estimation problem is reduced to a non-linear least squares (NLS) problem. For this reason (2.11) is also known as the NLS method.

The stochastic ML method [55] is obtained by considering the coefficients in $\boldsymbol{\alpha}$ as independent random variables. Specifically it is assumed that not only the noise in $\boldsymbol{w}$ but also the coefficients in $\boldsymbol{\alpha}$ follow a zero-mean complex Gaussian distribution. The parameters of the model are then $(\boldsymbol{f}, \boldsymbol{\gamma})$ and the stochastic ML estimator reads

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\gamma}}) = \arg\max_{\boldsymbol{f},\boldsymbol{\gamma}} \ln p(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{\gamma}) \tag{2.12}$$

$$= \arg\max_{\boldsymbol{f},\boldsymbol{\gamma}} \ln \int p(\boldsymbol{y}|\boldsymbol{\alpha}; \boldsymbol{f}) p(\boldsymbol{\alpha}; \boldsymbol{\gamma}) \mathrm{d}\boldsymbol{\alpha} \tag{2.13}$$

$$= \arg\min_{\boldsymbol{f},\boldsymbol{\gamma}} \ln |\boldsymbol{\Sigma}_{\boldsymbol{x}} + \sigma^2\mathbf{I}| + \boldsymbol{y}^{\mathrm{H}}(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \sigma^2\mathbf{I})^{-1}\boldsymbol{y}, \tag{2.14}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ takes the form in (2.3).

Both the deterministic ML and stochastic ML can be extended to the MMV case. We have not considered the details of those extensions here to limit the length of the text.

The algorithms that are derived in Paper A and B are extensions of the stochastic ML method and we therefore spend some time to discuss the properties of this class of methods (we do not wish to claim that the following properties are necessarily valid for our algorithms). We start by discussing the positive traits of the ML methods:

- The ML methods are known to be robust to violations of the statistical assumptions of white noise and no statistical correlation between the coefficients $\{\alpha_l\}$ [2, 24, 54, 56]. Note that this does not mean that they are optimal for these cases, just that their performance is not severely degraded when these assumptions do not hold.

- Both the deterministic and stochastic ML methods are known to be asymptotically consistent (estimate converges in probability to true value) and efficient (the estimator attains the Cramér-Rao bound) as N tends to infinity [2, 50, 57].

The asymptotic properties of the MMV extensions of the estimators has also been thoroughly studied. It turns out that the stochastic ML method is asymptotically consistent and efficient as M tends to infinity [58, 59] while deterministic ML possesses neither of these properties [50, 59]. This is due to the fact that as $M \to \infty$ the number of parameters which must be estimated by deterministic ML also goes to infinity (the matrix of coefficients $\mathbf{B}$ grows infinitely large).

For a given N and M it has been shown that the variances of the estimates determined by stochastic ML are upper bounded by the variances of the estimates obtained by deterministic ML [56]. In other words deterministic ML is statistically less efficient than stochastic ML.

- The ML methods do not require an estimate of the covariance matrix $\mathbf{\Sigma}_y$ as their input as is the case with subspace methods. That is a significant benefit in cases where only one or just a few measurement vectors are available.

The ML methods also have some negative traits:

- The objective functions in (2.11) and (2.14) are highly multimodal with many local minima [60]. This makes it hard to realize the ML estimators in practice. The usual approach is to devise a suboptimal initialization procedure which tries to find an estimate in the vicinity of the global minimizer. A search algorithm (for example gradient descent) is then used to refine the initial value. The estimation accuracy of such a procedure relies completely on the accuracy of the initialization and only convergence to a local minimizer can be guaranteed. See for example [61–63].
- As is also the case for the subspace methods, it is assumed that the number of sinusoids is known or estimated with another method.

2.3 Sparsity-based Line Spectral Estimation

When the number of sinusoids is unknown a priori the classical methods for LSE all require that another method is used to obtain an estimate of this number. Such a two-stage procedure is disadvantageous. The information theoretic approaches for model order estimation for example require that a set of frequencies is estimated for each considered model order [17], thus leading to large use of computational resources. To alleviate these issues the sparsity-based approach to LSE inherently estimates the number of sinusoids. A non-exhaustive list of prior work in this area is [5, 64–77].

2.3.1 Compressed Sensing of Line Spectra

The interest in sparse estimation within the signal processing community was revived in the mid-2000s with the appearance of *compressed sensing* [14–16]. By assuming a certain signal structure compressed sensing allows for a signal to be sampled at a rate below the Nyquist rate without any loss of information. The complete Nyquist-sampled signal can be recovered from the compressed measurements using sparse estimation techniques.

We here briefly give an overview of compressed sensing of line spectral signals. In compressed sensing the focus is on the recovery of the signal $\boldsymbol{x}$ from the observations $\boldsymbol{y}$, rather than on the estimation of the signal parameters $(L, \boldsymbol{f})$. There are three main cases considered in the literature:

1. *Compressed sensing of line spectra:* The recovery of the signal $\boldsymbol{x}$ from incomplete observation of $\boldsymbol{y}$ is considered in [78–80]. By incomplete we mean that either only a subset of the entries in $\boldsymbol{y}$ are available or that a low-dimensional compressed version of $\boldsymbol{x}$ is observable through a linear observation model. In the case of subsampled observation this task amounts to that of *interpolation* of the unobserved samples. These works answer questions like "What is the minimum number of samples required for reconstruction of the vector $\boldsymbol{x}$?"

2. *Denoising of line spectra:* The estimation of $\boldsymbol{x}$ from complete observation of $\boldsymbol{y}$ is considered in [81, 82]. If a noisy observation $\boldsymbol{y}$ of the line spectral signal $\boldsymbol{x}$ is observed, we can use sparse estimation methods to obtain an estimate of $\boldsymbol{x}$ by denoising $\boldsymbol{y}$. These works focus on questions like "What is the mean-square error of the denoised estimate of $\boldsymbol{x}$?"

3. *Super-resolution of line spectra:* The works [83–85] consider the *extrapolation* of the vector $\boldsymbol{x}$ form observations $\boldsymbol{y}$, i.e. the objective is to recover the signal samples outside the observed time interval. That can only realistically be achieved in very favorable conditions (frequencies are well separated, low noise). This type of works answer questions like "Considering a certain amount of samples to be extrapolated, what is the smallest error with which that extrapolation can be achieved?". The name super-resolution arises from the fact that extrapolation in the time domain corresponds to super-resolution of the fine details in the frequency-domain. Some of these works are also directly interested in the estimation of the frequencies $\{f_l\}$.

A common trait to all above approaches is their reliance on sparse regularization to perform the signal recovery. In many cases this signal recovery stage also estimates the signal parameters $(L, \boldsymbol{f})$. In this work we consider algorithms that can be applied in all of the above approaches.

2.3.2 The Grid Approximation

The use of sparse regularization is not directly applicable in the LSE setting because the frequencies are continuous-valued parameters. The first works in the field [5, 12, 13, 65, 74, 78, 86–88] therefore proceeded to quantize the vector $\boldsymbol{f}$ in the signal model (2.1) to obtain an approximated signal model

$$\boldsymbol{y} = \mathbf{A}(\bar{\boldsymbol{f}})\bar{\boldsymbol{\alpha}} + \boldsymbol{w}, \tag{2.15}$$

where $\bar{\boldsymbol{f}}$ is a vector containing a grid of frequency values on $[0, 1)$ and $\bar{\boldsymbol{\alpha}}$ is a sparse vector (it has only a few non-zero entries). The location of the non-zero entries in $\bar{\boldsymbol{\alpha}}$ then encodes the frequencies $\{f_l\}$. The length of vectors $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{\alpha}}$ is much larger than L and this length determines the accuracy of the signal model approximation. This approach reduces the LSE problem to a sparse estimation problem with a linear observation model ($\bar{\boldsymbol{f}}$ is fixed and known). Such a problem can for example be solved by ℓ_1-norm regularization (which gives the tightest convex relaxation of the ℓ_0-pseudo norm [89]) using the least absolute shrinkage and selection operator (LASSO),

$$\min_{\bar{\boldsymbol{\alpha}}} \|\bar{\boldsymbol{\alpha}}\|_1 + \lambda \|\boldsymbol{y} - \mathbf{A}(\bar{\boldsymbol{f}})\bar{\boldsymbol{\alpha}}\|_2^2, \tag{2.16}$$

where $\lambda > 0$ is a parameter for selecting the tradeoff between sparsity and fidelity of the signal recovery.

By solving (2.16) the number of sinusoids is inherently estimated (it is the number of non-zero entries in the solution for $\bar{\boldsymbol{\alpha}}$). However, the use of the approximate signal model (2.15) requires one to choose the number of points in the quantization grid $\bar{\boldsymbol{f}}$. Choosing this value too small leads to large approximation errors, also known as basis mismatch [90]. Choosing this value large implies that the dimensions of the problem (2.16) are large, leading to large computational complexity of its solution. With many grid points the columns of $\mathbf{A}(\bar{\boldsymbol{f}})$ also become highly correlated, making the solution of (2.16) numerically challenging. The presence of correlation also precludes a theoretical analysis of recovery guarantees using the conventional discrete compressed sensing setup.

2.3.3 Escaping the Grid

Due to the mentioned issues with the gridded approach there has been many works which seek to forego the restriction to a grid. Early attempts include grid refinement [64, 67] and interpolation methods [68, 69]. Later methods completely forego the use of a grid [70–72, 75–77, 79, 81–84]. In this thesis we work with two classes of methods for sparsity-based LSE which do not use a grid, namely atomic norm soft thresholding and a Bayesian approach. These two methods are shortly surveyed below and we refer to Papers B and C for details.

Atomic Norm Soft Thresholding

The atomic norm [79, 81–84, 91] provides a generalization of the ℓ_1-norm to continuous parameter spaces. When applied to LSE using a formulation similar to the LASSO its solution can be obtained by a semidefinite program,

$$\begin{array}{ll} \underset{v\in\mathbb{R},\, \boldsymbol{x}\in\mathbb{C}^N,\, \mathbf{T}\in\mathbb{C}^{N\times N}}{\text{minimize}} & \|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \tau v + \frac{2\tau}{N}\operatorname{tr}(\mathbf{T}) \\ \text{subject to} & \begin{pmatrix} \mathbf{T} & \boldsymbol{x} \\ \boldsymbol{x}^{\mathrm{H}} & v \end{pmatrix} \succeq 0, \\ & \mathbf{T} \text{ Hermitian Toeplitz,} \end{array} \tag{2.17}$$

where $\tau > 0$ is a regularization parameter. Recall that a Toeplitz matrix is one which has constant diagonals. The approach in (2.17) is known as atomic norm soft thresholding (AST) [81]. In Paper C a fast primal-dual interior point method for the solution of (2.17) is derived.

Bayesian Method

The Bayesian method proceeds as follows. We build an estimator which effectively estimates the number of sinusoids by starting with the model (2.1) where $L = L_{\max}$ is selected as an upper bound on the number of sinusoids to be estimated[3]. The estimator is then designed to disable those components which are not needed by selecting the corresponding coefficients or variances to be zero (i.e., $\alpha_l = 0$, $\gamma_l = 0$). Bayesian methods for sparse estimation provide us with a multitude of ways to obtain such behaviour using a sparsity-promoting prior on each α_l. Examples include the sparse Bayesian learning prior model [24, 72, 92–95] or the Bernoulli-Gaussian prior model [73, 96–99]. Not surprisingly, these Bayesian methods lead to optimization problems that share some terms with those of deterministic and stochastic ML in (2.11) and (2.14). For example, the evidence maximization [100] approach (also known as Type-II inference [101] in the sparse Bayesian learning literature) proceeds by marginalizing the coefficients $\boldsymbol{\alpha}$ to obtain a marginal posterior probability. Taking the maximum a-posteriori (MAP) estimate of $(\boldsymbol{f}, \boldsymbol{\gamma})$ amounts to solving the optimization problem

$$\begin{array}{ll} \underset{\boldsymbol{f}\in[0,1)^{L_{\max}},\, \boldsymbol{\gamma}\in\mathbb{R}^{L_{\max}}}{\text{minimize}} & \ln|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \sigma^2\mathbf{I}| + \boldsymbol{y}^{\mathrm{H}}(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \sigma^2\mathbf{I})^{-1}\boldsymbol{y} + g(\boldsymbol{\gamma}) \\ \text{subject to} & \gamma_l \geq 0 \text{ for } l = 0, \ldots, L_{\max}-1, \end{array} \tag{2.18}$$

[3] While we do not know the number of sinusoids, it is not unrealistic to assume that some upper bound is known. We can for example never expect to estimate more sinusoids that there are observations available and so $L_{\max} = N$ is an obvious choice.

where $\mathbf{\Sigma}_{\boldsymbol{y}}$ is as in (2.3) and $g(\boldsymbol{\gamma})$ is a regularization term[4] that promotes each entry of $\boldsymbol{\gamma}$ to be zero.

It is clear that the Bayesian approach endorses stochastic ML (2.14) with a regularization term on the effective number of sinusoids. It is then reasonable to think that (2.18) may inherit the good performance of stochastic ML. In the numerical evaluation in Paper B that is indeed shown to be the case. This does, however, also mean that the objective in (2.18) inherits the very non-convex and highly multimodal behaviour that makes the solution of stochastic ML hard to compute. In Paper B we propose an approach to solving (2.18) that is guaranteed to converge to a local minimum of the objective. Convergence to the global minimum, as is the case with the convex formulation (2.17), cannot be guaranteed. Despite of this we see in Paper B that the local solution of (2.18) (it may be that the global solution is found, but we have no guarantee of that) can outperform the global solution of (2.17) in terms of estimation accuracy.

2.3.4 Rank Minimization Relaxation as a Unifying Framework

To give a unified view of sparsity-based LSE methods we can view them as methods that estimate the covariance matrix of $\boldsymbol{x}$ via relaxations of rank-minimization problems. The rank minimization problems of interest are known to be NP-hard (meaning that no known polynomial time algorithm can solve them) [102, 103]. The rank term is therefore replaced with some proxy and the choice of this proxy yields different algorithms for LSE.

First note that the entries of the signal vector $\boldsymbol{x}$ (2.1) constitute a wide-sense stationary process. It then follows that the covariance matrix $\mathbf{\Sigma}_{\boldsymbol{x}}$ has Toeplitz structure. Since $\mathbf{\Sigma}_{\boldsymbol{x}}$ is a covariance matrix it is also Hermitian. We need these properties in the following.

Atomic Norm Soft Thresholding

Consider the AST method in (2.17) and note that using the pseudo Schur complement [104] it can be rewritten in the form

$$\begin{array}{ll} \underset{\boldsymbol{x}\in\mathbb{C}^N,\mathbf{T}\in\mathbb{C}^{N\times N}}{\text{minimize}} & \|\boldsymbol{x}-\boldsymbol{y}\|_2^2+\tau\boldsymbol{x}^{\mathrm{H}}\mathbf{T}^{+}\boldsymbol{x}+\frac{2\tau}{N}\operatorname{tr}(\mathbf{T}) \\ \text{subject to} & \mathbf{T}\succeq 0, \\ & \mathbf{T} \text{ Hermitian Toeplitz}, \\ & \boldsymbol{x}\in\mathcal{R}(\mathbf{T}), \end{array} \tag{2.19}$$

[4] For the Bernoulli-Gaussian prior model used in Paper B, this regularization term reads $g(\boldsymbol{\gamma}) = \sum_{l=0}^{L_{\max}-1} \mathbf{1}[\gamma_l = 0]\ln\zeta + \mathbf{1}[\gamma_l \neq 0]\ln(1-\zeta)$, where $\zeta \in [0, 1/2]$ is a regularization parameter and $\mathbf{1}[\cdot]$ is the indicator function. Note that a change of variables is required to obtain (2.18) from the optimization problem formulated in the paper.

where $\mathbf{T}^+$ is the Moore-Penrose pseudo inverse of $\mathbf{T}$ and $\mathcal{R}(\mathbf{T})$ is the column space of the matrix $\mathbf{T}$. The matrix $\mathbf{T}$ in this optimization problem is an estimate of the signal covariance matrix $\mathbf{\Sigma}_{\boldsymbol{x}}$. Let's decipher each term of the problem (2.19):

- The first two constraints guarantee that $\mathbf{T}$ is a valid covariance matrix.
- The first term of the objective $\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$ ensures that the estimate of the signal vector $\boldsymbol{x}$ is consistent with the observation $\boldsymbol{y}$. This term therefore handles the noise.
- The second term $\tau \boldsymbol{x}^{\mathrm{H}} \mathbf{T}^+ \boldsymbol{x}$ together with the constraint $\boldsymbol{x} \in \mathcal{R}(\mathbf{T})$ expresses the "fit" between the signal vector $\boldsymbol{x}$ and the covariance matrix $\mathbf{T}$. Specifically it is the data dependent term of the negative log likelihood function.[5] The log-determinant term of the log likelihood is not included because the size of the entries of the covariance matrix is regularized by the last term in (2.19).
- Since $\mathbf{T}$ is Hermitian the term $\mathrm{tr}(\mathbf{T})$ is equal to the nuclear norm of $\mathbf{T}$, which is the tightest convex relaxation of $\mathrm{rank}(\mathbf{T})$ [91, 106, 107]. There is a strong relation to ℓ_1-norm minimization which is seen by letting $\boldsymbol{\lambda}$ be a vector containing the eigenvalues of $\mathbf{T}$. Then $\|\boldsymbol{\lambda}\|_1 = \mathrm{tr}(\mathbf{T})$ is the tightest convex relaxation of $\|\boldsymbol{\lambda}\|_0 = \mathrm{rank}(\mathbf{T})$ [89]. This is not surprising since the atomic norm is a generalization of the ℓ_1-norm to continuous parameter spaces.

It is clear that AST estimates a signal vector $\boldsymbol{x}$ which is consistent with the data along with a low-rank Hermitian Toeplitz matrix $\mathbf{T}$ that is a likely covariance matrix of the estimated $\boldsymbol{x}$. For more details on the low-rank viewpoint we refer to [108], albeit that work considers the (simpler) case of noise-free observations.

Bayesian Method

To interpret the Bayesian methods embodied in (2.18) as a rank minimization method we need a result known as the Carathéodory parameterization [2, 109, 110] of a covariance matrix. It states that for any positive semidefinite Hermitian Toeplitz matrix $\mathbf{T} \in \mathbb{C}^{N \times N}$ of rank $L < N$ there exists unique[6]

[5] Recall that $\boldsymbol{x}$ is a zero-mean degenerate complex Gaussian random vector with covariance $\mathbf{\Sigma}_{\boldsymbol{x}}$. The probability density function of such a random vector is defined on $\boldsymbol{x} \in \mathcal{R}(\mathbf{\Sigma}_{\boldsymbol{x}})$ and reads [105, pp. 527–528]

$$p(\boldsymbol{x}) = (\pi^N |\mathbf{\Sigma}_{\boldsymbol{x}}|_+)^{-1} \exp(-\boldsymbol{x}^{\mathrm{H}} \mathbf{\Sigma}_{\boldsymbol{x}}^+ \boldsymbol{x}),$$

where $|\cdot|_+$ is the pseudo-determinant (product of non-zero eigenvalues).

[6] Uniqueness up to permutations of the entries in $\boldsymbol{f}$ and $\boldsymbol{\gamma}$.

values $\boldsymbol{f} \in [0,1)^L$ and $\boldsymbol{\gamma} \in (0,\infty)^L$ such that

$$\mathbf{T} = \mathbf{A}(\boldsymbol{f})\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}(\boldsymbol{f}), \tag{2.20}$$

where $\boldsymbol{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma}) \in \mathbb{R}^{L\times L}$ and $\mathbf{A}(\boldsymbol{f}) \in \mathbb{C}^{N\times L}$ is as described in the text following (2.1).

We also know that the converse is true: For any choice of $\boldsymbol{f} \in [0,1)^L$ and $\boldsymbol{\gamma} \in [0,\infty)^L$ the matrix in (2.20) is positive semidefinite Hermitian Toeplitz. If the entries in $\boldsymbol{f}$ are distinct, then the rank of $\mathbf{T}$ is equal to the number of non-zero entries in $\boldsymbol{\gamma}$.

Let's assume that $L_{\max} < N$ in (2.18)[7] Then there is a one-to-one correspondence between $(\boldsymbol{f},\boldsymbol{\gamma})$ and the Hermitian Toeplitz matrix $\mathbf{T}$. That allows us to reparameterize (2.18) in the form

$$\begin{array}{ll} \underset{\mathbf{T}\in\mathbb{C}^{N\times N}}{\text{minimize}} & \ln|\mathbf{T}+\sigma^2\mathbf{I}| + \boldsymbol{y}^{\mathrm{H}}(\mathbf{T}+\sigma^2\mathbf{I})^{-1}\boldsymbol{y} + \tilde{g}(\mathbf{T}) \\ \text{subject to} & \mathbf{T} \succeq 0, \\ & \mathbf{T} \text{ Hermitian Toeplitz}, \\ & \operatorname{rank}(\mathbf{T}) \leq L_{\max}, \end{array} \tag{2.21}$$

where $\tilde{g}(\mathbf{T}) = g(\boldsymbol{\gamma})$ with $\boldsymbol{\gamma}$ obtained from the Carathéodory parameterization of $\mathbf{T}$ in (2.20)[8].

Deciphering this optimization problem, we note that the first two constraints guarantee that $\mathbf{T}$ is a valid covariance matrix of $\boldsymbol{x}$. The first two terms of the objective function are the negative log-likelihood for observing $\boldsymbol{y}$ if $\boldsymbol{x}$ has covariance matrix $\mathbf{T}$. The last term $\tilde{g}(\mathbf{T})$ regularizes the rank. We do not expect the constraint $\operatorname{rank}(\mathbf{T}) \leq L_{\max}$ to affect the solution because $L_{\max}$ by assumption is an upper bound on the number of sinusoids to be estimated.

Despite the very different derivations that was used to obtain AST (2.19) and the Bayesian method (2.21), we can now directly relate the two methods. They are both composed of two types of terms. The first type of terms enforces consistency between the covariance $\mathbf{T}$ and the observed signal. These terms also account for the observation noise. The second type of terms is the rank regularization terms. The Bayesian method employs a non-convex rank regularizer $\tilde{g}(\mathbf{T})$ while AST employs a convex relaxation of the rank term, namely the nuclear norm $\operatorname{tr}(\mathbf{T})$.

We note that it is possible to rewrite the problem (2.21) into a form that has even stronger resemblance to the problem (2.19) (some terms of the two problems become equal). That is done by considering a certain Type-I or dual

[7] In Papers A and B we consider $L_{\max} = N$, but selecting for example $L_{\max} = N-1$ would not make any difference in practice. We assume $L_{\max} < N$ as it greatly simplifies the analysis here. Without this assumption the parameterization (2.20) is not unique.

[8] For the Bernoulli-Gaussian prior model used in Paper B (i.e., with $g(\cdot)$ mentioned in footnote 4) this regularization term reads $\tilde{g}(\boldsymbol{\gamma}) = \operatorname{rank}(\mathbf{T})\ln\zeta + (L_{\max} - \operatorname{rank}(\mathbf{T}))\ln(1-\zeta)$. Clearly this term promotes low rank when $\zeta < \frac{1}{2}$.

space formulation [111–113]; see also [77]. To retain focus on the main issue, we have not pursued this avenue here.

Other Methods

The above view also allows us to propose any number of new sparsity-based LSE methods by choosing other rank regularizers or other ways to enforce consistency with the observed data. That is indeed what the reweighted atomic norm minimization approach [114] amounts to.

Even the classical methods for LSE can be seen as estimation of a low-rank Toeplitz covariance matrix, where the model order selection method plays the role of the rank regularization term. As an example, for a given model order, we can think of the MUSIC algorithm (see Sec. 2.2.1) as an approach to find a Hermitian Toeplitz covariance matrix that has column space approximately equal to the signal subspace.

In the following chapter we see how the rank minimization viewpoint can provide some insights into the application of LSE algorithms for parametric channel estimation.

Chapter 3

Application to Wireless Channel Estimation

3.1 A Simple Model of Wireless Communication

As an example application of LSE, we consider the parametric approach to wireless channel estimation. That is also the subject of Paper A. In the paper the main focus is on algorithm design, specifically the design of an iterative receiver implementing parametric channel estimation. Here we expand on the assumptions made regarding the channel model and the relation to LSE.

Consider a simple model of a wireless communication system as depicted in Fig. 3.1, where a single transmitter emits a signal intended to a single receiver. Typically the radio waves propagate from the transmitter to the receiver via multiple paths. Along these paths the waves may experience reflection, diffraction and scattering when interacting with physical objects along their way. At the receiver a superposition of contributions from each of the propagation paths is observed. The process which takes as input the signal emitted from the transmitter and gives as output the superposition of signals that can be observed at the receiver is known as the wireless channel.

The wireless channel distorts the signal. In order to effectively communicate information over the wireless channel, this distortion must be compensated for. That is most commonly done by first estimating the effect of the wireless channel followed by a correction stage (equalization). We focus on the channel estimation stage and employ a parametric model to do so.

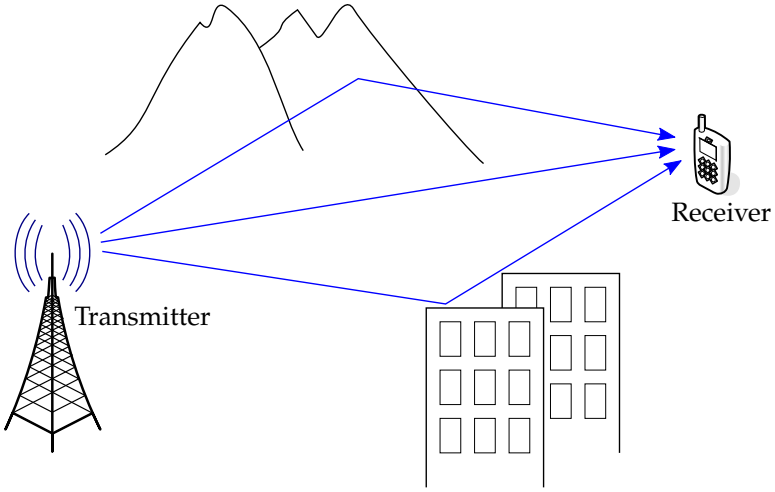

Fig. 3.1: A wireless communication system with multiple propagation paths.

3.2 Signal Model of an OFDM System

This section contains a few passages that have been reused from Paper A with minor or no modifications.

To discuss the channel estimation task in detail the signal model of a typical communication system is now introduced. An orthogonal frequency-division multiplexing (OFDM) system is considered. OFDM is interesting to study because a majority of modern communication standards are based on it. The baseband signal model for such a system is derived below, making strong (but common) assumptions regarding the behaviour of the transmitter and receiver. Essentially it is assumed that the communication system has ideal behaviour in all regards which do not directly relate to channel estimation. These assumptions are made to allow us to focus solely on the aspects related to channel estimation.

Consider a single OFDM symbol duration in a system with N subcarriers. The (coded) bits to be transmitted are modulated to obtain the complex symbols $x_0, \ldots, x_{N-1}$ and the baseband processor in the transmitter emits

$$s(t) = \begin{cases} \sum_{n=0}^{N-1} x_n \exp(j2\pi\Delta_f n t) & t \in [-T_{\text{CP}}; T_{\text{sym}}], \\ 0 & \text{otherwise}, \end{cases} \tag{3.1}$$

where Δ_f gives the subcarrier spacing, $T_{\text{sym}} = \Delta_f^{-1}$ is the OFDM symbol duration and T_{CP} is the cyclic prefix length.

All filtering effects in the transmitter and receiver radio frequency (RF) front-ends are modelled together by a linear time-invariant (LTI) filter with

impulse response $c(\tau)$. The wireless channel is assumed to be constant for the duration of an OFDM symbol, an assumption known as *block fading*. The wireless channel is also modelled as an LTI filter with impulse response $h(\tau)$. A common assumption in OFDM system design is that the compound response of these two filters has excess delay no larger than T_{CP}, i.e., that

$$(c * h)(\tau) = 0 \quad \text{for} \quad \tau \notin [0; T_{\mathrm{CP}}], \tag{3.2}$$

where $*$ denotes convolution. The signal at the output of the receiver RF front-end then reads[1]

$$r(t) = (c * h * s)(t) + w(t), \tag{3.3}$$

where $w(t)$ is a noise process. The receiver samples $r(t)$, removes the cyclic prefix and calculates the discrete Fourier transform to obtain the observed vector $\tilde{\boldsymbol{y}}$. It can be shown [115] that

$$\tilde{\boldsymbol{y}} = \mathbf{X}\mathbf{C}\boldsymbol{h} + \tilde{\boldsymbol{w}}, \tag{3.4}$$

where $\tilde{\boldsymbol{w}}$ is a noise vector and $\mathbf{X}$ contains the symbols $\{x_n\}$ on the diagonal and zeros elsewhere. The matrix $\mathbf{C} = \mathrm{diag}(\boldsymbol{c})$ is diagonal with the entries of $\boldsymbol{c}$ on the diagonal. The vectors $\boldsymbol{h}$ and $\boldsymbol{c}$ contain the frequency response of the filters $h(\tau)$ and $c(\tau)$ at the subcarrier frequencies. Specifically, the entries of $\boldsymbol{h}$ are samples of the (continuous) Fourier transform $H(f)$ of $h(\tau)$:

$$h_n = H(n\Delta_f) = \int_0^{T_{\mathrm{sym}}} h(\tau) \exp(-j2\pi\Delta_f n\tau)\, \mathrm{d}\tau, \qquad n = 0, \ldots, N-1. \tag{3.5}$$

A similar definition holds for $\boldsymbol{c}$.

It is commonly assumed that the RF front-ends have constant response within the system bandwidth, such that $\mathbf{C} = \mathbf{I}$. This is supported by the fact that some guard subcarriers are left unused at the edge of the transmission bandwidth, thus lessening the design criteria of the bandlimiting filters in the transmitter and receiver [115]. Here we make the weaker assumption that $\mathbf{C}$ is known a priori. It is not currently clear in the literature if the assumption $\mathbf{C} = \mathbf{I}$ is appropriate when designing systems with parametric channel estimation and, if not, how an estimate of $\mathbf{C}$ should be obtained. We envision that it could be obtained through a calibration procedure, but this question falls outside the scope of this work.

The receiver in an OFDM system makes inference about the information communicated through $\{x_n\}$ from the observation $\tilde{\boldsymbol{y}}$. In order to do so, the channel vector $\boldsymbol{h}$ must be estimated – a process called channel estimation. To

[1] We ignore numerous imperfections which exist and must be handled in wireless communication systems, such as (inter-symbol, inter-carrier, inter-cell) interference and (time-, phase-, frequency-) synchronization offsets, etc.

facilitate estimation of $\boldsymbol{h}$ a subset of the symbols $\{x_n\}$ are assigned as *pilots*. This means that they are known a priori by both the transmitter and receiver. Let $\mathcal{P}$ denote the indices of the pilots and let $v_{\mathcal{P}}$ denote the vector composed of the entries in vector v with indices in $\mathcal{P}$. The channel estimator takes as input the noisy observations of the compound channel frequency response at the pilot subcarriers,

$$\boldsymbol{y} = (\mathbf{X}^{-1}\mathbf{C}^{-1}\tilde{\boldsymbol{y}})_{\mathcal{P}} = \boldsymbol{h}_{\mathcal{P}} + \boldsymbol{w}_{\mathcal{P}}, \tag{3.6}$$

where $\boldsymbol{w} = \mathbf{X}^{-1}\mathbf{C}^{-1}\tilde{\boldsymbol{w}}$. We make the assumption that $\boldsymbol{w}$ is a vector of white, zero-mean complex Gaussian noise with entrywise variance σ^2. That is a reasonable assumption when both the pilot symbols (that are found in $\mathbf{X}$) and the entries on the diagonal of $\mathbf{C}$ have approximately constant magnitude across the subcarriers.

3.3 Linear Minimum Mean-Square Error Channel Estimation

The usual approach in wireless receivers is to use a linear minimum mean-square error (LMMSE) estimator of $\boldsymbol{h}$ based on $\boldsymbol{y}$. Assuming that $\boldsymbol{h}$ has zero mean (as is the case in practically all channel models), the LMMSE estimator reads

$$\hat{\boldsymbol{h}}_{\text{LMMSE}} = \boldsymbol{\Sigma}_{\cdot,\mathcal{P}} \left(\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{P}} + \sigma^2\mathbf{I}\right)^{-1} \boldsymbol{y}, \tag{3.7}$$

where $\boldsymbol{\Sigma} = \mathbb{E}\left[\boldsymbol{h}\boldsymbol{h}^{\text{H}}\right]$ is the covariance matrix of $\boldsymbol{h}$. Here $\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{P}}$ denotes rows and columns selected according to the index set $\mathcal{P}$ and "$\cdot$" denotes all rows or columns.

Such an LMMSE estimator essentially interpolates the remaining entries of $\boldsymbol{h}$ from noisy observations of $\boldsymbol{h}_{\mathcal{P}}$. It can meaningfully do so if a Nyquist sampling criterion[2] is fulfilled, that is when the Fourier transform of $\boldsymbol{h}$ has limited support. The assumption of excess delay limited to T_{CP} in (3.2) provides such a Nyquist sampling criterion. If the indices in $\mathcal{P}$ are equispaced and differ by Δ_P the Nyquist criterion is fulfilled when

$$\Delta_f \Delta_P T_{\text{CP}} < 1. \tag{3.8}$$

Many communication standards operate far from this boundary to increase resilience to noise. As an example in the 3GPP LTE specification [116] the criterion is $\Delta_P < 12.8$ in normal operation, but the system uses $\Delta_P = 6$.

[2]The role of time and frequency is swapped from the usual definition of the Nyquist sampling theorem.

The estimation accuracy of the LMMSE estimator depends crucially on the assumed covariance matrix $\mathbf{\Sigma}$. The choice of this covariance matrix is not straightforward as statistical information about the current propagation channel must be available to calculate it. We now discuss how it can be estimated using parametric channel estimation.

3.4 Parametric Channel Estimation

To characterize the channel covariance matrix we start with a fairly general stochastic model of the wireless channel. In this model the impulse response $h(\tau)$ is drawn from a zero-mean stochastic process with autocorrelation function

$$R(\tau, \tau') = \rho(\tau)\delta(\tau - \tau'). \tag{3.9}$$

Channels of this form are usually referred to as *uncorrelated scattering* channels. The function $\rho(\tau)$ is known as the *power-delay profile* (PDP) and it specifies the second-order statistics of the channel impulse response.

Recall that the channel vector $\boldsymbol{h}$ contains samples of the Fourier transform $H(f)$ of $h(\tau)$. It is fairly easy to show that (3.9) implies that $H(f)$ is a wide-sense stationary random process. This in turn implies that the channel covariance matrix $\mathbf{\Sigma} = \mathbb{E}\left[\boldsymbol{h}\boldsymbol{h}^{\mathrm{H}}\right]$ is Toeplitz. For simplicity it is assumed in the following that $\boldsymbol{h}$ has a complex Gaussian distribution.

We now assume that $\mathbf{\Sigma}$ has low rank[3] (or at least that it is well approximated by a low rank matrix). Edfors *et al.* [117] make the same assumption. We discuss it in further detail in Sec. 3.5. The parametric channel estimator proceeds as informally described here:

1. *Find a low-rank Toeplitz covariance matrix $\mathbf{\Sigma}$ that is consistent with the observation $\boldsymbol{y}$.*

2. *Use this covariance matrix to obtain an estimate of the channel vector $\boldsymbol{h}$.*

3.4.1 Solution via Line Spectral Estimation

Recall from Sec. 2.3.4 that LSE algorithms actually solve the problem in Step 1 of the parametric channel estimator. This link explains the name "parametric". We find the connection between Step 1 and LSE to be very important and now explore it in more detail.

Denote the rank of $\mathbf{\Sigma}$ as L and assume $L < N$ (that follows from the low-rank assumption). Then according to the Caratheódory parameterization of

[3]This implies that we allow the distribution of $\boldsymbol{h}$ to be *degenerate*, i.e., the set of possible outcomes of $\boldsymbol{h}$ is a subspace of $\mathbf{C}^N$.

the channel covariance matrix, it can be uniquely decomposed as (see (2.20))[4]

$$\mathbf{\Sigma} = \mathbf{\Psi}(\boldsymbol{\tau})\mathbf{\Gamma}\mathbf{\Psi}^{\mathrm{H}}(\boldsymbol{\tau}), \tag{3.10}$$

where the matrix $\mathbf{\Psi}(\boldsymbol{\tau}) \in \mathbb{C}^{N\times L}$ has entries $(\mathbf{\Psi}(\boldsymbol{\tau}))_{n,l} = \exp(-j2\pi\Delta_f n\tau_l)$ for $n = 0,\ldots,N-1$, $l = 0,\ldots,L-1$. The number L denotes the number of *virtual multipath components*. The vector $\boldsymbol{\tau} \in \mathbb{R}^L$ contains the *virtual multipath delays* and $\mathbf{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ where $\boldsymbol{\gamma} \in \mathbb{R}^L$ is a vector of *virtual multipath powers*. Note that $0 \leq \tau_l < \Delta_f^{-1}$ and $\gamma_l > 0$ for $l = 0,\ldots,L-1$.

The above implies that there is a one-to-one mapping between the Toeplitz matrix $\mathbf{\Sigma}$ and the set of parameters $(L, \boldsymbol{\tau}, \boldsymbol{\gamma})$. So we can replace estimation of the low-rank Toeplitz covariance matrix with estimation of $(L, \boldsymbol{\tau}, \boldsymbol{\gamma})$. It also means that $\boldsymbol{h}$ lies in the subspace spanned by the columns of $\mathbf{\Psi}(\boldsymbol{\tau})$ and it can be written as

$$\boldsymbol{h} = \mathbf{\Psi}(\boldsymbol{\tau})\boldsymbol{\alpha}. \tag{3.11}$$

for some $\boldsymbol{\alpha} \in \mathbb{C}^L$. In fact, $\boldsymbol{\alpha}$ is a zero-mean random vector with covariance matrix $\mathbf{\Gamma}$.

The equation (3.11) shows that $\boldsymbol{h}$ is a line spectral signal. Recall that our observations are of the form (3.6). Inserting (3.11) into (3.6) we get

$$\boldsymbol{y} = \mathbf{\Psi}_{\mathcal{P},\cdot}(\boldsymbol{\tau})\boldsymbol{\alpha} + \boldsymbol{w}_{\mathcal{P}}. \tag{3.12}$$

The estimation of $\boldsymbol{\tau}$ and $\boldsymbol{\gamma}$ from $\boldsymbol{y}$ is then an LSE problem, cf. (2.1).

Denote the obtained estimates as $\hat{L}$, $\hat{\boldsymbol{\tau}}$ and $\hat{\boldsymbol{\gamma}}$. The channel vector can then be estimated in (at least) two different ways: *1)* by inserting $(\hat{L}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\gamma}})$ into (3.10) to obtain an estimate of the channel covariance matrix $\mathbf{\Sigma}$. This estimate can then be used in the LMMSE estimator (3.7); *2)* by using the estimated variances $\hat{\boldsymbol{\gamma}}$ and the model (3.12) to obtain the LMMSE estimate[5] $\hat{\boldsymbol{\alpha}}$ of $\boldsymbol{\alpha}$. Insert the estimate $\hat{\boldsymbol{\alpha}}$ into (3.11). Since the LMMSE estimator commutes over the linear transformation $\mathbf{\Psi}(\hat{\boldsymbol{\tau}})$ [118, p. 349] these two approaches give the same result, namely

$$\hat{\boldsymbol{h}}_{\mathrm{param}} = \mathbf{\Psi}(\hat{\boldsymbol{\tau}}) \left(\mathbf{\Psi}_{\mathcal{P},\cdot}^{\mathrm{H}}(\hat{\boldsymbol{\tau}})\mathbf{\Psi}_{\mathcal{P},\cdot}(\hat{\boldsymbol{\tau}}) + \sigma^2\hat{\mathbf{\Gamma}}^{-1} \right)^{-1} \mathbf{\Psi}_{\mathcal{P},\cdot}^{\mathrm{H}}(\hat{\boldsymbol{\tau}})\boldsymbol{y}. \tag{3.13}$$

This also make the link between parametric and LMMSE channel estimation clear: The former is a method to obtain a covariance matrix for use in the latter.

[4] We have reparameterized the matrix $\mathbf{A}(f)$ in (2.20) to the form $\mathbf{\Psi}(\boldsymbol{\tau})$ as it simplifies the relation to the model (3.14) introduced in Sec. 3.4.2

[5] The majority of the literature on parametric channel estimation uses a least-squares estimate of the coefficients in $\boldsymbol{\alpha}$. We here instead use the LMMSE estimator as it provides a natural link to (3.7). This difference in approach from the literature is minor and has no practical effect.

3.4.2 Detour: The Usual Model

In the previous section we obtained a parametric channel estimator by assuming that the channel covariance matrix $\mathbf{\Sigma}$ is low-rank. In the literature on parametric channel estimation [11, 13, 119, 120] and also in Paper A the parametric channel model is derived from another assumption. Specifically it is assumed that the wireless channel is composed of multiple distinct propagation paths, each with their own multipath delay and multipath coefficient. Such a model follows intuitively when we think of the wireless channel as depicted in Fig. 3.1. This is mathematically modelled by the following parametric description of the channel impulse response:

$$h(\tau) = \sum_{l=0}^{\tilde{L}-1} \tilde{\alpha}_l \delta(\tau - \tilde{\tau}_l), \tag{3.14}$$

Here, $\tilde{L}$ is the number of multipath components. The lth multipath coefficient is denoted as $\tilde{\alpha}_l \in \mathbb{C}$ with corresponding multipath delay $\tilde{\tau}_l \in \mathbb{R}$. It is assumed that $\tilde{\alpha}_l$ is zero-mean complex Gaussian with variance $\tilde{\gamma}_l$ and that $\tilde{\alpha}_l$ is independent of $\tilde{\alpha}_k$ for $k \neq l$. This model fulfills the uncorrelated scattering assumption and it is a special case of the model (3.9).

Taking the Fourier transform (3.5) of (3.14) we recover a model for $\boldsymbol{h}$ of the form (3.11) (with $(\boldsymbol{\tau}, \boldsymbol{\alpha})$ replaced by $(\tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\alpha}})$). Under this model the vector $\boldsymbol{h}$ is also a zero-mean complex Gaussian vector with covariance matrix having the form in (3.10) (with $(\boldsymbol{\tau}, \boldsymbol{\gamma})$ replaced by $(\tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\gamma}})$).

By identifying the values $(\tilde{L}, \tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\alpha}})$ with $(L, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ it is seen that we can obtain the parametric channel estimator by either assuming the low-rank model associated with (3.9) or the model in (3.14). The former is actually a generalization of the latter and it encompasses a much wider class of channels (the model (3.14) is obtained by letting $\rho(\tau)$ be a sum of Dirac delta functions in (3.9)). We have found it to be much more instructive to think in terms of the low-rank model due to the following shortcoming of the model (3.14).

Cluster Model of the Channel Impulse Response

If parametric channel estimation is obtained from (3.14) the number of parameters to be estimated is proportional to $\tilde{L}$ and it is assumed that $\tilde{L}$ is small (relative to N). There are a few examples of channels where such a criterion may be fulfilled, for example in underwater acoustic communications [121]. It is, however, broadly accepted that such a criterion is not fulfilled for wireless channels in general [11, 119, 122]. This means that the model (3.14) in many cases is invalid in practice. That issue is often circumvented with the argument that each of the terms in the sum (3.14) actually represents a *cluster* of multipath components [11, 119]. In this view each cluster consists of

a large number of paths with approximately the same delay. The number $\tilde{L}$ then signifies the number of clusters and it may realistically be small.

The argumentation based on clusters has some shortcomings. It is not clear how similar the delays of the multipath components must be for them to belong to the same cluster. If this "cluster width" is selected too large, the use of a Dirac delta function in (3.14) becomes increasingly inaccurate, while a smaller cluster width leads to larger values of $\tilde{L}$. As a result it is not clear which value $\tilde{L}$ takes even if a particular realization of the wireless channel impulse response is provided.

Benefits of the Low-rank Viewpoint

Conversely, with the low-rank approach to obtain the parametric channel estimator we do not need to make such arguments based on clusters. In this view the rank of $\mathbf{\Sigma}$ directly gives the channel degrees of freedom (DoF). The DoF is oftentimes introduced with heuristic arguments based on the channel impulse response (see [11, 123]). In many ways the low-rank viewpoint gives a rigorous approach that can replace the heuristic description of the channel DoF via a cluster view of the model (3.14). In those cases where the wireless channel is composed of such clusters, the channel covariance matrix is indeed low rank for all practical purposes, as demonstrated in Sec. 3.5.

We also note that by considering the parametric channel estimator as an approach to estimate a low-rank covariance matrix, the estimates $(\hat{L}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\gamma}})$ can be interpreted as the parameters of a virtual channel impulse response of the form (3.14) which is valid within the bandwidth of the communication system. That follows immediately from the decomposition (3.10).

3.5 The Rank of the Channel Covariance Matrix

We still have not discussed whether the channel covariance matrix ever has low rank and, if so, under which propagation conditions this is the case. Edfors *et al.* [117] show that it is (under their modelling assumptions) and we here conduct a numerical study to investigate some further details.

3.5.1 Setup

The channel covariance matrix and the PDP are intimately connected: The channel covariance matrix is Hermitian Toeplitz with the autocorrelation sequence of the channel vector $\boldsymbol{h}$ as the first row (the first row specifies the whole covariance matrix). This autocorrelation sequence is obtained by sampling the Fourier transform of the PDP. To discuss the properties of the channel covariance matrix we must be very specific with respect to which area (of movement of transmitter, receiver and other objects in the environment)

the expectation that appears in the PDP is taken over. That is important because it affects the amount of variation that is seen in the (temporal) dispersion profile of the channel. A clear specification of which type of PDP is considered is not generally seen in the literature, something we find to be a significant shortcoming. We consider three different specifications of the PDP, depending on how strong statistical assumptions are made:

- *Robust PDP* [117]: This is the most general specification of the PDP which makes the weakest assumptions regarding the channel. Specifically, in the spirit of (3.2), it is only assumed that the PDP is constant on the interval $[0, T_{\mathrm{CP}}]$ and zero outside this interval. This PDP is useful when the receiver has no knowledge about the propagation environment. One may say that there are no limitations on the area over which the entities (transmitter, receiver and other objects in the environment) may move.

- *Global PDP* [124]: Here it is assumed that the general propagation environment is known. It could for example be known that the propagation environment is an urban macro cell. The entities may move freely within this environment. If the channel model takes the form (3.14), this corresponds to stochastic coefficients $\{\alpha_l\}$ and delays $\{\tau_l\}$. In other words objects move such that the delay incidence pattern changes.

- *Local PDP* [124]: We finally consider a local PDP, where the entities move over such small areas, that the delay incidence pattern is unchanged. This means that the radius of movement is much smaller than the distance the radio waves travel in the timespan of one delay resolution bin. These radii must thus be much smaller than $\frac{c}{N\Delta_f}$, where c is the speed of light. If the channel model takes the form (3.14), this corresponds to stochastic coefficients $\{\alpha_l\}$ and fixed, deterministic delays $\{\tau_l\}$. The covariance matrix estimated by the parametric channel estimator corresponds to a local PDP.

To gain some insight into the behaviour of the channel covariance matrix, we perform a numerical investigation of its rank under two different propagation scenarios with different channel models. These are the same scenarios that are considered in Paper A and we refer to that paper for further details. Here is a short summary of the channel models and system parameters:

- *Scenario A*: Uses the ITU-R M.2135 UMa NLOS channel model [125]. An OFDM system with $N = 1024$ subcarriers, subcarrier spacing $\Delta_f = 25\,\mathrm{kHz}$ and cyclic prefix length $T_{\mathrm{CP}} = 5200\,\mathrm{ns}$ is considered.

- *Scenario B*: Uses the IEEE 802.15.a Outdoor NLOS channel model [126]. An OFDM system with $N = 1024$ subcarriers, subcarrier spacing $\Delta_f = 250\,\mathrm{kHz}$ and cyclic prefix length $T_{\mathrm{CP}} = 800\,\mathrm{ns}$ is considered.

While both of these scenarios use a channel model of the form (3.14), only that in Scenario A has relatively small L. In Scenario B the value of L is so large that we cannot hope to estimate each of the multipath components with the parametric channel estimator.

3.5.2 Power-Delay Profiles and Eigenvalues

In Figs. 3.2 and 3.3 two realizations of the local PDP are shown alongside the global and robust PDPs, for Scenario A and B, respectively. As a first observation, note how the effective support of the PDP (interval of delays where it has significant power) becomes smaller as stronger statistical assumptions are made. This means that the channel effectively has few DoFs and it is this property we seek to exploit to increase channel estimation accuracy.

From each of the PDPs a corresponding frequency-domain autocorrelation function can be calculated (using the Wiener-Khinchin theorem). Forming a Toeplitz matrix from the autocorrelation function gives the corresponding channel covariance matrix. We denote the covariance matrices corresponding to the three PDP assumptions as $\mathbf{\Sigma}_{\text{robust}}$, $\mathbf{\Sigma}_{\text{global}}$ and $\mathbf{\Sigma}_{\text{local}}$. Fig. 3.4 depicts the sorted eigenvalues of these matrices in both Scenario A and Scenario B. Only up to the first 150 (Scenario A) or 250 (Scenario B) eigenvalues are shown out of the total number $N = 1024$ of eigenvalues. Notice how most of the eigenvalues are very small. This is again an indication that the channel vector $\boldsymbol{h}$ effectively lies in a low-dimensional space.

It is also clear that while most of the eigenvalues are small, they are not exactly zero (this not due to numerical precision). We therefore define a method to neglect small eigenvalues which do not represent significant signal power. For that purpose the eigenvalues are normalized such that their average value is 1. The eigenvalues below 10^{-4} are then set to zero. This means that whenever the SNR is significantly below 40 dB, the removed signal power is neglectable in comparison to the noise power. Even in perfect conditions wireless communication systems practically always operate significantly below 40 dB SNR. For that reason we find this to be a conservative approach to thresholding the eigenvalues. Taking the resulting number of non-zero eigenvalues gives an effective rank. In the following we refer to this effective rank simply as the rank.

3.5.3 Rank and Mean-Square Error

The distribution of the rank of $\mathbf{\Sigma}_{\text{local}}$ is investigated in Fig. 3.5 for both Scenario A and B. It is presented as the empirical cumulative distribution function (CDF) obtained based on multiple realizations of the local PDP. In Table 3.1 the corresponding ranks are given for the channel covariance matrices $\mathbf{\Sigma}_{\text{robust}}$ and $\mathbf{\Sigma}_{\text{global}}$.

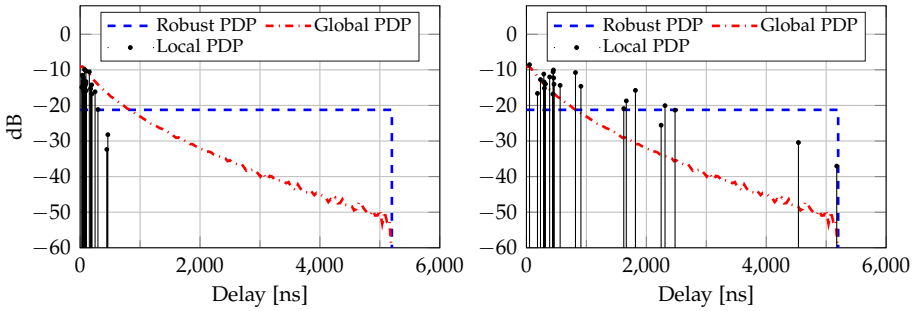

Fig. 3.2: Power-delay profiles in Scenario A. The local PDPs are obtained by drawing the number of multipath components, delays and multipath coefficient powers from the channel model. Two realizations are shown (left and right). The global PDP is obtained by averaging 50,000 realizations of the local PDP. The global and robust PDPs are the same in both plots.

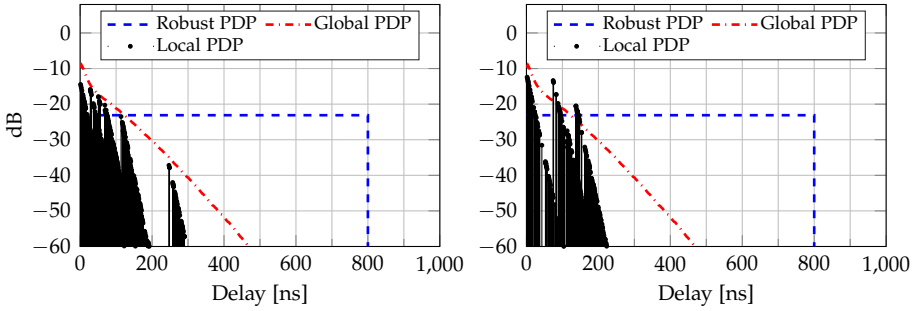

Fig. 3.3: Power-delay profiles in Scenario B. The local PDPs are obtained by drawing the number of multipath components, delays and multipath coefficient powers from the channel model. Two realizations are shown (left and right). The global PDP is obtained by averaging 50,000 realizations of the local PDP. The global and robust PDPs are the same in both plots.

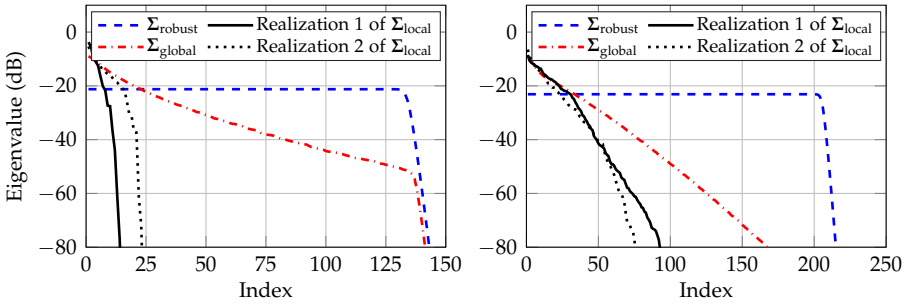

Fig. 3.4: Sorted eigenvalues of the channel covariance matrices in Scenario A (left) and Scenario B (right). The two realizations of $\mathbf{\Sigma}_{\text{local}}$ are computed from the local PDPs depicted in Figs. 3.2 and 3.3.

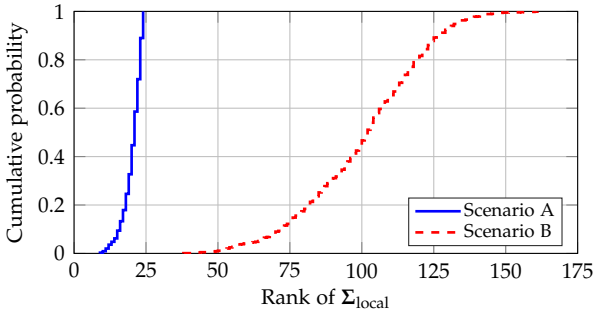

Fig. 3.5: Empirical CDFs of rank of $\mathbf{\Sigma}_{\text{local}}$ when a threshold is applied to the eigenvalues. The plot is obtained by estimating the CDF from $1,000$ realizations of the local PDP in each of the scenarios.

	Rank of $\mathbf{\Sigma}_{\text{robust}}$	Rank of $\mathbf{\Sigma}_{\text{global}}$	$N\Delta_f T_{\text{CP}}$
Scenario A	141	139	133
Scenario B	213	147	205

Table 3.1: Rank of the covariance matrices generated from the robust and global PDPs. The smallest eigenvalues are set to zero such that up to -40 dB of the signal power is removed.

We first note that the rank of $\mathbf{\Sigma}_{\text{robust}}$ is very close to $N\Delta_f T_{\text{CP}}$. This fact is clear from the following intuition: $(N\Delta_f)^{-1}$ is the width of the delay resolution bins associated to the periodogram. The number $N\Delta_f T_{\text{CP}}$ then gives how many delay resolution bins the robust PDP is spanning over. This is approximately equal to the channel DoF associated to the robust PDP and therefore roughly coincides with the rank of $\mathbf{\Sigma}_{\text{robust}}$.

The rank is a measure of the channel DoF. Recall that the parametric channel estimator effectively is an approach for obtaining a covariance matrix for use in a LMMSE channel estimator. The rank of this covariance matrix gives the dimension of a subspace of which the channel vector $\boldsymbol{h}$ is assumed to be an element. The lower the dimension of this subspace is, the better estimation accuracy can be expected. The parametric channel estimator estimates $\mathbf{\Sigma}_{\text{local}}$ and we wish to compare this estimator to an LMMSE estimator which knowns either $\mathbf{\Sigma}_{\text{global}}$ or $\mathbf{\Sigma}_{\text{robust}}$.

To do that we depict in Fig. 3.6 the normalized mean-square error (NMSE) that is achieved by the LMMSE estimator of $\boldsymbol{h}$ (3.7) when using different covariance matrices. The NMSE depicted in Fig. 3.6 is obtained as a Monte Carlo simulation over channel realizations. In each trial a realization of $\mathbf{\Sigma}_{\text{local}}$ is obtained from the channel model. In the channel models we use here it follows that $\boldsymbol{h}$ is a zero-mean complex Gaussian random vector with covariance matrix $\mathbf{\Sigma}_{\text{local}}$. Let's say that noisy observations at the pilot subcarriers

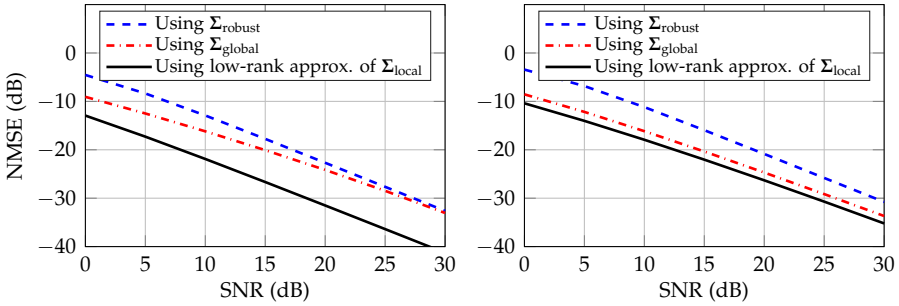

Fig. 3.6: NMSE of the estimator (3.7) when using different channel covariance matrices. Both Scenario A (left) and Scenario B (right) are considered. Results are averaged over 1,000 realizations of $\mathbf{\Sigma}_{\text{local}}$. See the text for details.

are available of the form (3.6). The noise variance σ^2 is selected according to the SNR under consideration with the signal power given by the average of the diagonal of $\mathbf{\Sigma}_{\text{local}}$. The set of pilot subcarrier indices $\mathcal{P}$ is selected to be equispaced with $\Delta_P = 4$. The LMMSE estimator of the vector $\boldsymbol{h}$ (both data and pilot subcarriers) using different covariance matrices is then considered. Specifically, three estimators are obtained by inserting $\mathbf{\Sigma}_{\text{robust}}$, $\mathbf{\Sigma}_{\text{global}}$ and the rank-truncated version of $\mathbf{\Sigma}_{\text{local}}$ into (3.7). The NMSE of each of these three estimators can be calculated in closed form based on the true covariance matrix $\mathbf{\Sigma}_{\text{local}}$ (which has not been rank truncated). The normalization is relative to the signal power given by the trace of $\mathbf{\Sigma}_{\text{local}}$. The NMSE values depicted in Fig. 3.6 are averaged over 1,000 realizations of $\mathbf{\Sigma}_{\text{local}}$.

We first consider Scenario A. In this case $\mathbf{\Sigma}_{\text{local}}$ has significantly lower rank than both $\mathbf{\Sigma}_{\text{global}}$ and $\mathbf{\Sigma}_{\text{robust}}$ (Fig. 3.5 and Table 3.1). This is not surprising since the channel model in Scenario A is truly specular, i.e., it has the form (3.14) with L relatively small (Fig. 3.2). This means that we expect the parametric channel estimator to perform significantly better than the LMMSE estimator, which is also confirmed by Fig. 3.6 and the simulation results in Paper A.

In Scenario B the rank of $\mathbf{\Sigma}_{\text{local}}$ is only marginally smaller than the rank of $\mathbf{\Sigma}_{\text{global}}$ (Fig. 3.5 and Table 3.1) and so we do not expect that a parametric channel estimator can give a significant improvement in estimation accuracy over an LMMSE estimator which knows $\mathbf{\Sigma}_{\text{global}}$. However, an improvement over an LMMSE estimator which uses $\mathbf{\Sigma}_{\text{robust}}$ is expected. These statements are again confirmed by Fig. 3.6 and the simulation results in Paper A.

3.5.4 Summary

In conclusion, we can see that the above formalism based on the rank of the channel covariance matrix gives a good understanding of the applica-

bility and behaviour of the parametric channel estimator in comparison to an LMMSE estimator. It is clear that the prior knowledge available to the LMMSE estimator through the covariance matrix is of crucial importance to its performance.

Clearly we have assumed that the parametric channel estimator can perfectly estimate $\mathbf{\Sigma}_{\text{local}}$. That is not the case in practice and so the above considerations are to be understood as the best possible performance one can obtain with a parametric channel estimator. From the simulation results of Paper A it seems that the parametric channel estimator is able to obtain a reasonably accurate estimate of $\mathbf{\Sigma}_{\text{local}}$ though.

The above discussion is based on a numerical investigation of the covariance matrices which are generated from two different channel models. Transposition of the drawn conclusions to real channels thus stands on the assumption that these models accurately reflect the behaviour of the rank of the covariance matrix rank of these real channels. That is not necessarily the case because the channel models have not been devised with this purpose in mind. It would be interesting to investigate the channel covariance matrix and its rank directly based on a set of measurements, but that falls outside the scope of this work.

Chapter 4

Contributions & Outlook

4.1 Contributions

The overall theme of this thesis is the design of algorithms for sparsity-based LSE. We pay particular attention to the design of algorithms that are practically applicable. This for example means that the algorithms we design inherently can estimate nuisance parameters such as the noise variance. It also means that there is a strong focus on reducing the computational complexity of the algorithms. We further demonstrate how sparsity-based LSE algorithms can be incorporated into larger inference frameworks by considering their use for parametric channel estimation in an iterative receiver design.

In this section we give a short summary of each of the papers in the thesis and the main contributions and findings of the papers are discussed.

Paper A: In view of the discussion in Chapter 3, it is clear that the parametric approach provides a way to obtain channel estimators with high estimation accuracy. Another very powerful approach to increasing channel estimation accuracy is via joint channel estimation and decoding schemes. In this paper the simultaneous application of these two ideas for wireless receiver design is investigated. Special attention is paid to rigorous methodologies for the design of such receiver algorithms. The Bayesian method for sparsity-based LSE is used to obtain the parametric channel estimator. It is shown that a mean-field implementation of this channel estimator can naturally be merged with belief propagation decoding in a variational Bayesian inference framework.

The proposed receiver algorithm is assessed using an extensive numerical evaluation. It is shown that the proposed algorithm indeed increases the channel estimation accuracy in comparison to a selection of state-of-the-art joint channel estimation and decoding algorithms. It is also demonstrated that the proposed receiver design can have lower bit-error rate (BER) than the

baseline algorithms in scenarios with high SNR and large modulation orders. Further, the proposed design may allow for the number of pilot signals to be lowered (compared to the baseline algorithms) without incurring an increase in BER.

Paper B: In the application of LSE for parametric channel estimation in a receiver (Paper A) it is observed that the high computational complexity of the algorithm precludes it from implementation in an actual wireless communication system. Motivated by this limitation we study the computational complexity of LSE algorithms in a general setting in Paper B. The Bayesian method for sparsity-based LSE is investigated and it is shown that an algorithm with low computational complexity can be obtained by appropriately formulating an inference scheme using the Bernoulli-Gaussian prior model. At the core of the algorithm lies a so-called superfast method (specifically the generalized Schur algorithm) for decomposing the inverse of a Hermitian Toeplitz covariance matrix along with a non-uniform fast Fourier transform.

The asymptotic per-iteration computational complexity of the obtained algorithm scales as $\mathcal{O}(N \log^2 N)$ in the problem size N. That is a significant improvement over the $\mathcal{O}(N^3)$ scaling of most of the state-of-the-art algorithms. The proposed algorithm is also shown to outperform the selected baseline algorithms in terms successful recovery of the line spectral frequencies.

Paper C: In this paper we consider the numerical solution of the convex optimization problem that appears in atomic norm soft thresholding (AST). Not surprisingly, this optimization problem has some of the same Toeplitz structure that is exploited in the superfast LSE algorithm of Paper B. We investigate how this structure can be exploited to obtain a fast algorithm for solving AST.

Since AST is a convex optimization problem this work takes quite a different approach to algorithm design than Paper B. Specifically we propose a novel primal-dual interior point method (IPM) to solve this problem. AST is usually formulated as a semidefinite programming problem, but that form precludes the derivation of a fast primal-dual IPM due to the presence of $\mathcal{O}(N^2)$ dual variables. We show that by reformulating AST as a non-symmetric conic programming problem only $\mathcal{O}(N)$ dual variables are needed. The problem structure in the conic formulation depends crucially on a non-self-dual convex cone. The derivation of the primal-dual IPM involves an analysis of the dual of this cone. Specifically we show how membership of the dual cone can approximately be determined in a fast manner.

A numerical investigation shows that the proposed primal-dual IPM accurately solves AST and that it does so much faster than any other known method.

4.2 Outlook

Algorithmic Aspects

This thesis provides a selection of highly accurate sparsity-based LSE algorithms that have low computational complexity. In particular, they are able to tackle very large estimation problems with only small execution times. The superfast LSE algorithm in Paper B can for example handle a signal of length $N = 16,384$ in less than 10 s on an average laptop. This significant reduction in estimation time when compared to state-of-the-art methods extends the scope of problems for which it is practically feasible to employ sparsity-based LSE.

From an algorithmic viewpoint there are a number of interesting extensions. For the FastAST algorithm (Paper C) the extensions to incomplete data [79] or multiple measurement vectors [127] appear fairly straightforward (the superfast LSE algorithm of Paper B already incorporate these cases). We also expect that the techniques used to obtain FastAST can be fruitfully applied to a number of optimization problems involving related conic constraints, for example those listed in [128].

For both of the algorithms in Papers B and C a non-trivial extension is to the multi-dimensional harmonic retrieval problem [129]. This extension is non-trivial because the Toeplitz structure is replaced by a block-Toeplitz structure. The work [130] may provide some helpful insights in this regard.

In Wireless Communication

While it is our hope that these fast algorithms will open applications in a wide array of fields, we here mention a few examples in wireless communication. We have discussed the use of LSE algorithms for channel estimation. In a search for higher data transmission rates, wireless systems tend to utilize wider bandwidths of spectrum [131, 132]. For parametric channel estimators that trend translates into LSE problems of larger dimensions that must be solved. Wireless receivers are embedded real-time systems and only very limited computational resources are available for the channel estimation task. The reduction in computational requirements is therefore crucial to the practical use of parametric channel estimators. The algorithms we have developed are therefore an important step towards the adoption of LSE in this application.

We have introduced a low-rank formalism for understanding the applicability of parametric channel estimation in channels that are not purely specular. The use of this formalism and its implications for channel modelling and estimation is open to further enquiry.

Throughout the thesis we have used as a running example the application

to parametric estimation of the delay-domain dispersion profile of a wireless channel. We have not discussed the related application to parametric estimation of the angular-domain dispersion profile that arises when multi-antenna systems are considered. The algorithms and the low-rank formalism of this thesis have a similar application here. The low-complexity algorithms may prove important in this application when considering so-called massive-MIMO systems that are envisioned to employ a very large number of antennas [133–135] (albeit there is a long way to $N = 16,384$ antennas).

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

[2] P. Stoica and R. L. Moses, *Spectral analysis of signals*. Prentice Hall, 2005.

[3] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing*. McGraw-Hill Boston, 2000.

[4] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[5] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug 2005.

[6] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Process.*, vol. 40, pp. 590–600, Mar. 1992.

[7] R. Carriere and R. L. Moses, "High resolution radar target modeling using a modified Prony estimator," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 13–18, Jan. 1992.

[8] A. Jakobsson, A. L. Swindlehurst, and P. Stoica, "Subspace-based estimation of time delays and Doppler shifts," *IEEE Trans. Signal Process.*, vol. 46, no. 9, pp. 2472–2483, Sep. 1998.

[9] U. Tureli, H. Liu, and M. D. Zoltowski, "OFDM blind carrier offset estimation: ESPRIT," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1459–1461, Sep. 2000.

[10] X. Andrade, J. N. Sanders, and A. Aspuru-Guzik, "Application of compressed sensing to the simulation of atomic systems," *Proc. Nat. Academy of Sciences*, vol. 109, no. 35, pp. 13 928–13 933, Jul. 2012.

[11] W. Bajwa, A. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, Jun. 2010.

[12] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.

[13] N. L. Pedersen, C. N. Manchón, D. Shutin, and B. H. Fleury, "Application of Bayesian hierarchical prior modeling to sparse channel estimation," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 3487–3492.

[14] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[15] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.

[16] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[17] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, Jul. 2004.

[18] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 387–392, Apr. 1985.

[19] E. Fishler and H. V. Poor, "Estimation of the number of sources in unbalanced arrays via information theoretic criteria," *IEEE Trans. Signal Process.*, vol. 53, no. 9, pp. 3543–3553, Aug. 2005.

[20] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2006–2021, Nov. 2010.

[21] Y. I. Abramovich, N. K. Spencer, and A. Y. Gorokhov, "Detection-estimation of more uncorrelated Gaussian sources than sensors in nonuniform linear antenna arrays. I. Fully augmentable arrays," *IEEE Trans. Signal Process.*, vol. 49, no. 5, pp. 959–971, May 2001.

[22] H.-T. Wu, J.-F. Yang, and F.-K. Chen, "Source number estimators using transformed Gerschgorin radii," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1325–1333, Jun. 1995.

[23] H. Linhart and W. Zucchini, *Model selection*, ser. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1986.

[24] D. Wipf and S. Nagarajan, "Beamforming using the relevance vector machine," in *Proc. 24th Int. Conf. Mach. Learning*, Jun. 2007, pp. 1023–1030.

[25] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[26] Y. Li and Y. Chi, "Off-the-grid line spectrum denoising and estimation with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1257–1269, Mar. 2016.

[27] J. M. Kim, O. K. Lee, and J. C. Ye, "Compressive MUSIC: Revisiting the link between compressive sensing and array signal processing," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 278–301, Jan. 2012.

[28] L. Marple, "Resolution of conventional Fourier, autoregressive, and special ARMA methods of spectrum analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, May 1977, pp. 74–77.

[29] S. T. Smith, "Statistical resolution limits and the complexified Cramér-Rao bound," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1597–1609, May 2005.

[30] B. G. R. De Prony, "Essai éxperimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, a différentes températures," *J. de l'école polytechnique*, vol. 1, no. 22, pp. 24–76, 1795.

[31] P. Stoica, T. Soderstrom, and F. Ti, "Asymptotic properties of the high-order Yule-Walker estimates of sinusoidal frequencies," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 11, pp. 1721–1734, Nov. 1989.

[32] Y. Chan and R. Langford, "Spectral estimation via the high-order yule-walker equations," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 5, pp. 689–698, Oct. 1982.

[33] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 5, pp. 814–824, May 1990.

[34] T. K. Sarkar and O. Pereira, "Using the matrix pencil method to estimate the parameters of a sum of complex exponentials," *IEEE Antennas Propag. Mag.*, vol. 37, no. 1, pp. 48–55, Feb. 1995.

[35] J. P. Burg, "Maximum entropy spectral analysis," in *Proc. Annu. Int. Meeting, Soc. of Explor. Geophys.*, Oct. 1967.

[36] J. Ables, "Maximum entropy spectral analysis," *Astronomy Astrophysics Suppl. Series*, vol. 15, p. 383, Jun. 1974.

[37] A. Papoulis, "Maximum entropy and spectral estimation: A review," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1176–1186, Dec. 1981.

[38] M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *IEEE Trans. Signal Process.*, vol. 39, no. 11, pp. 2436–2449, Nov. 1991.

[39] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1110–1121, May 1991.

[40] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[41] J. Li and P. Stoica, "An adaptive filtering approach to spectral estimation and SAR imaging," *IEEE Trans. Signal Process.*, vol. 44, no. 6, pp. 1469–1484, Jun. 1996.

[42] P. Stoica, A. Jakobsson, and J. Li, "Matched-filter bank interpretation of some spectral estimators," *Signal Process.*, vol. 66, no. 1, pp. 45–59, Apr. 1998.

[43] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.

[44] P. Stoica, "List of references on spectral line analysis," *Signal Process.*, vol. 31, no. 3, pp. 329–340, Apr. 1993.

[45] V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophysical J. Int.*, vol. 33, no. 3, pp. 347–366, Sep. 1973.

[46] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[47] R. Roy and T. Kailath, "ESPRIT – estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, pp. 984–995, Jul. 1989.

[48] S.-Y. Kung, K. S. Arun, and B. D. Rao, "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem," *J. the Optical Soc. of America*, vol. 73, pp. 1799–1811, Dec. 1983.

[49] R. Kumaresan and D. W. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. Aerosp. Electron. Syst.*, no. 1, pp. 134–139, Jan. 1983.

[50] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramér-Rao bound," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 5, pp. 720–741, May 1989.

[51] J. E. Evans, J. R. Johnson, and D. F. Sun, "Application of advanced signal processing techniques to angle of arrival estimation in ATC navigation and surveillance systems," Lincoln Laboratory, MIT, USA, Tech. Rep., 1982.

[52] B. D. Rao and K. V. S. Hari, "Weighted subspace methods and spatial smoothing: analysis and comparison," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 788–803, Feb. 1993.

[53] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 806–811, Apr. 1985.

[54] P. Stoica and A. Nehorai, "Statistical analysis of two nonlinear least-squares estimators of sine-wave parameters in the colored-noise case," *Circuits, Syst., Signal Process.*, vol. 8, no. 1, pp. 3–15, Mar. 1989.

[55] J. Bohme, "Estimation of source parameters by maximum likelihood and nonlinear regression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 9, Mar. 1984, pp. 271–274.

[56] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1783–1795, Oct. 1990.

[57] M. Viberg, B. Ottersten, and A. Nehorai, "Estimation accuracy of maximum likelihood direction finding using large arrays," in *Proc. 25th Asilomar Conf. Signals, Syst., Computers*, Nov. 1991, pp. 928–932.

[58] P. Stoica, B. Ottersten, M. Viberg, and R. L. Moses, "Maximum likelihood array processing for stochastic coherent sources," *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 96–105, Jan. 1996.

[59] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample ML techniques for parameter estimation and detection in array processing," in *Radar Array Process. (Chap. 4)*, S. Haykin, Ed. Springer-Verlag, 1993.

[60] P. Stoica, R. L. Moses, B. Friedlander, and T. Soderstrom, "Maximum likelihood estimation of the parameters of multiple sinusoids from noisy measurements," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 378–392, Mar. 1989.

[61] R. Kumaresan, L. Scharf, and A. Shaw, "An algorithm for pole-zero modeling and spectral analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 3, pp. 637–640, Jun. 1986.

[62] Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1081–1089, Oct. 1986.

[63] M. I. Miller and D. R. Fuhrmann, "Maximum-likelihood narrow-band direction finding and the EM algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 9, pp. 1560–1577, Sep. 1990.

[64] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug. 2005.

[65] A. Fannjiang and W. Liao, "Coherence pattern-guided compressive sensing with unresolved grids," *SIAM J. Imaging Sci.*, vol. 5, pp. 179–202, Feb. 2012.

[66] ——, "Super-resolution by compressive sensing algorithms," in *Proc. 46th Annu. Asilomar Conf. Signals, Syst., Computers*, Nov. 2012, pp. 411–415.

[67] G. Z. Karabulut, T. Kurt, and A. Yongaçoglu, "Estimation of directions of arrival by matching pursuit (EDAMP)," *EURASIP J. Wireless Commun. Networking*, pp. 197–205, Apr. 2005.

[68] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "Recovery of sparse translation-invariant signals with continuous basis pursuit," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4735–4744, Oct. 2011.

[69] K. Fyhn, F. Marco, S. Holdt *et al.*, "Compressive parameter estimation for sparse translation-invariant signals using polar interpolation," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 870–881, Feb. 2015.

[70] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3809–3822, Jul. 2012.

[71] L. Hu, J. Zhou, Z. Shi, and Q. Fu, "A fast and accurate reconstruction algorithm for compressed sensing of complex sinusoids," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5744 – 5754, Nov. 2013.

[72] T. L. Hansen, M.-A. Badiu, B. H. Fleury, and B. D. Rao, "A sparse Bayesian learning algorithm with dictionary parameter estimation," in *Proc. 8th IEEE Sensor Array and Multichannel Signal Process. Workshop*, Jun. 2014, pp. 385–388.

[73] M. A. Badiu, T. L. Hansen, and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Trans. Signal Process.*, vol. 65, pp. 2247–2261, May 2017.

[74] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, "Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6257–6261, Dec. 2011.

[75] D. Shutin, W. Wang, and T. Jost, "Incremental sparse Bayesian learning for parameter estimation of superimposed signals," in *Proc. 10th Int. Conf. Sampling Theory Applicat.*, Jul. 2013, pp. 6–9.

[76] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse Bayesian inference," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 38 – 43, Jan. 2011.

[77] Z. Yang and L. Xie, "On gridless sparse methods for line spectral estimation from complete and incomplete data," *IEEE Trans. Signal Process.*, vol. 63, pp. 3139–3153, Jun. 2015.

[78] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Appl. Computational Harmonic Anal.*, vol. 35, no. 1, pp. 111–129, Jul. 2013.

[79] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7465–7490, Nov. 2013.

[80] E. J. Candes, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Computational Harmonic Anal.*, vol. 31, no. 1, pp. 59–73, Jul. 2011.

[81] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, Dec. 2013.

[82] G. Tang, B. N. Bhaskar, and B. Recht, "Near minimax line spectral estimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 499–512, Jan. 2015.

[83] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *J. Fourier Anal. Applicat.*, vol. 19, no. 6, pp. 1229–1254, Dec. 2013.

[84] ——, "Super-resolution from noisy data," *J. Fourier Anal. Applicat.*, vol. 19, no. 6, pp. 1229–1254, Dec. 2013.

[85] D. L. Donoho, "Superresolution via sparsity constraints," *SIAM J. Math. Anal.*, vol. 23, no. 5, pp. 1309–1331, Sep. 1992.

[86] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 374–377, Mar. 2002.

[87] M. A. Davenport, D. Needell, and M. B. Wakin, "Signal space CoSaMP for sparse recovery with redundant dictionaries," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6820–6829, Oct. 2013.

[88] P. Schniter, "Belief-propagation-based joint channel estimation and decoding for spectrally efficient communication over unknown sparse channels," *Physical Communication*, vol. 5, no. 2, pp. 91–101, Jun. 2012.

[89] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the k-support norm," in *Proc. Advances in Neural Inform. Process. Syst.*, 2012, pp. 1457–1465.

[90] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2182–2195, May 2011.

[91] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Math.*, vol. 12, no. 6, pp. 805–849, Dec. 2012.

[92] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[93] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learning Research*, pp. 211–244, Jun. 2001.

[94] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. 9th Int. Workshop Artificial Intell., Stat.*, Jan. 2003, pp. 3–6.

[95] N. L. Pedersen, C. N. Manchón, M.-A. Badiu, D. Shutin, and B. H. Fleury, "Sparse estimation using Bayesian hierarchical prior modeling for real and complex linear models," *Signal Process.*, vol. 115, pp. 94–109, Oct. 2015.

[96] J. J. Kormylo and J. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 482–488, May 1982.

[97] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[98] F. Dublanchet, J. Idier, and P. Duwaut, "Direction-of-arrival and frequency estimation using Poisson-Gaussian modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, Apr. 1997, pp. 3501–3504.

[99] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. Int. Symp. Inform. Theory Proc.*, Jul. 2011, pp. 2168–2172.

[100] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[101] J. O. Berger, *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[102] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Prog.*, vol. 128, no. 1-2, pp. 321–353, Jun. 2011.

[103] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," in *Proc. Amer. Control Conf.*, vol. 4, Jun. 2004, pp. 3273–3278.

[104] J. Gallier, "The Schur complement and symmetric positive semidefinite (and definite) matrices," *Penn Eng.*, Dec. 2010.

[105] C. R. Rao, *Linear statistical inference and its applications*, ser. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1973.

[106] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford University, 2002.

[107] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, Aug. 2010.

[108] Z. Yang and L. Xie, "Exact joint sparse frequency recovery via optimization methods," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5145–5157, Jun. 2014.

[109] N. D. Sidiropoulos, "Generalizing Caratheodory's uniqueness of harmonic parameterization to N dimensions," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1687–1690, May 2001.

[110] C. Carathéodory and L. Fejér, "Über den zusammenhang der extremen von harmonischen funktionen mit ihren koeffizienten und über den picard-landau'schen satz," *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 32, no. 1, pp. 218–239, Jan. 1911.

[111] Y. Wu and D. P. Wipf, "Dual-space analysis of the sparse linear model," in *Proc. Advances in Neural Inform. Process. Syst.*, 2012, pp. 1745–1753.

[112] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Proc. Advances in Neural Inform. Process. Syst.*, 2008, pp. 1625–1632.

[113] D. P. Wipf, B. D. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6236–6255, Sep. 2011.

[114] Z. Yang and L. Xie, "Enhancing sparsity and resolution via reweighted atomic norm minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 995–1006, Feb. 2016.

[115] M. Luise, R. Reggiannini, and G. M. Vitetta, "Blind equalization/detection for OFDM signals over frequency-selective channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1568–1578, Oct. 1998.

[116] *Evolved Universal Terrestrial Radio Access (E-UTRA), Physical channels and modulation*, 3GPP Std. 36.211, Rev. 12.8.0, Dec. 2015.

[117] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Börjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.

[118] S. M. Kay, *Fundamentals of statistical signal processing, volume I: estimation theory*. Prentice Hall, 1993.

[119] P. Schniter, "A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels," *IEEE J. Select. Topics in Signal Process.*, vol. 5, no. 8, pp. 1462–1474, Dec. 2011.

[120] G. Taubock and F. Hlawatsch, "A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2008, pp. 2885–2888.

[121] C. Berger, S. Zhou, J. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708 – 1721, Mar. 2010.

[122] G. L. Turin, "Communication through noisy, random-multipath channels," Ph.D. dissertation, Massachusetts Institute of Technology, 1956.

[123] W. U. Bajwa, A. Sayeed, and R. Nowak, "Sparse multipath channels: Modeling and estimation," in *Proc. Digital Signal Process., Signal Process. Educ. Workshop*, Jan. 2009, pp. 320–325.

[124] R. Heddergott, U. P. Bernhard, and B. H. Fleury, "Stochastic radio channel model for advanced indoor mobile communication systems," in *Proc. 8th IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun.*, Sep. 1997, pp. 140–144.

[125] *Guidelines for evaluation of radio interface technologies for IMT-Advanced*, ITU-R Std. M.2135-1, Dec. 2008.

[126] A. F. Molisch, D. Cassioli, C.-C. Chong, S. Emami, A. Fort, B. Kannan, J. Karedal, J. Kunisch, H. G. Schantz, K. Siwiak *et al.*, "A comprehensive standardized model for ultrawideband propagation channels," *IEEE Trans. Antennas Propag.*, vol. 54, no. 11, pp. 3151–3166, Nov. 2006.

[127] Y. Li and Y. Chi, "Off-the-grid line spectrum denoising and estimation with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1257–1269, Mar. 2016.

[128] B. Alkire and L. Vandenberghe, "Convex optimization problems involving finite autocorrelation sequences," *Math. Prog.*, vol. 93, no. 3, pp. 331–359, Dec. 2002.

[129] Y. Chi and Y. Chen, "Compressive two-dimensional harmonic retrieval via atomic norm minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1030–1042, Feb. 2015.

[130] Z. Yang, L. Xie, and P. Stoica, "Vandermonde decomposition of multilevel Toeplitz matrices with application to multidimensional super-resolution," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3685–3701, Jun. 2016.

[131] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology," *IEEE Wireless Com.*, vol. 17, no. 3, Jun. 2010.

[132] G. Yuan, X. Zhang, W. Wang, and Y. Yang, "Carrier aggregation for LTE-advanced mobile communication systems," *IEEE Commun. Mag.*, vol. 48, no. 2, Feb. 2010.

[133] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[134] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[135] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

Part II

Papers

Paper A

A Turbo Receiver for OFDM With Sparsity-Based Parametric Channel Estimation

Thomas L. Hansen, Peter B. Jørgensen, Mihai-Alin Badiu and Bernard H. Fleury

The paper has been submitted to the
IEEE Transactions on Signal Processing

The layout has been revised.

Abstract

In this work we design a receiver based on the turbo principle in the sense that it iterates between channel estimation and data decoding. The receiver incorporates sparsity-based parametric channel estimation. State-of-the-art sparsity-based turbo receivers simplify the channel estimation problem by restricting the multipath delays to a grid. Our receiver does not impose such a restriction. As a result it does not suffer from the leakage effect, which destroys sparsity. Communication at near capacity rates in high SNR requires a large modulation order. Due to the close proximity of modulation symbols in such systems, the grid-based approximation is of insufficient accuracy. We show numerically that a state-of-the-art turbo receiver with grid-based sparse channel estimation exhibits a bit error rate floor in the high SNR regime. On the contrary, our receiver performs very close to the "perfect channel state information bound" for all SNR values. We also demonstrate that our receiver design works well in diffuse channels, i.e., when the number of multipath components is large and each individual component cannot be resolved.

A.1 Introduction

Achieving high data-rate wireless communication with large spectral efficiency requires the use of higher-order modulation formats, e.g. up to 256-QAM in 3GPP LTE [1]. Clearly using a high modulation order presuppose a large signal-to-noise ratio (SNR), which will be supported by the envisioned transition to small-cell operation. The availability of channel estimation schemes which achieve high accuracy is crucial for receivers of systems with large modulation order operating in the high-SNR regime.

To facilitate channel estimation, current systems embed pilot symbols into the transmitted signal. In orthogonal frequency-division multiplexing (OFDM) systems, a number of subcarriers are assigned to transmit pilot symbols. The number of pilots is chosen to optimize throughput as a trade-off between the amount of bandwidth and power allocated to pilot transmission and fidelity of the channel estimate.

In this work we seek to improve upon this trade-off by designing a highly accurate channel estimator while requiring a low pilot overhead. We propose a unified receiver design that incorporates two main ideas: *a)* turbo architecture and *b)* sparsity-based parametric channel estimation.

A.1.1 Turbo Receiver Design

Classical receiver design employs a functional splitting of the process in the receiver into independent subtasks, as illustrated in Fig. A.1. Such a structure is suboptimal, since the information learned from the received signal in any

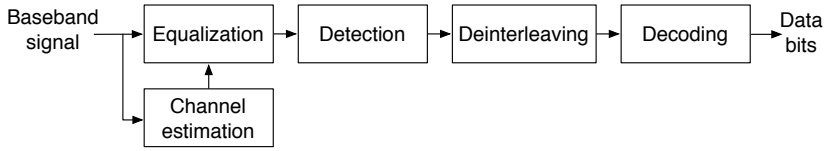

Fig. A.1: Flowchart of classical receiver design.

of the subtasks is only utilized in subsequent subtasks. To remedy this suboptimality feedback loops can be introduced between the functional blocks in the receiver. This approach is known as the turbo principle [2, 3] due to its resemblance to iterative decoding of turbo codes.

Application of the turbo principle has lead to many iterative receiver designs, e.g. [2–4]. Common to these works is that each of the subtasks are designed independently using traditional methods such as maximum likelihood (ML), maximum a-posteriori probability (MAP) or minimum mean squared error (MMSE). The work [5] introduced receiver design from the perspective of inference in a factor graph. This allows for the receiver subtasks to be designed *jointly* with a certain objective in mind; a common example is to seek the MAP estimate of the information bits. Due to tractability and computational constraints, approximate inference methods must be employed for turbo receiver design. Examples of such methods are expectation propagation [6], belief propagation (BP) with approximated messages [7], combined BP and mean-field (MF) [8, 9], relaxed BP [10] and generalized approximate message-passing (GAMP) [11].

A.1.2 Parametric Channel Estimation

The compound channel impulse response (CIR) is often modelled as a sum of distinct multipath components:

$$h(\tau) = \sum_{l=1}^{L} \alpha_l c(\tau - \tau_l), \tag{A.1}$$

where $c(\tau)$ denotes the combined impulse response of the filters in the transmitter and receiver RF front-ends. Here, L is the number of multipath components. The lth channel multipath coefficient is denoted as $\alpha_l \in \mathbb{C}$ with corresponding multipath delay $\tau_l \in \mathbb{R}$. We refer to propagation channels with a small number of multipath components L as specular. In this case the model (A.1) is parsimonious and it is advantageous to perform channel estimation by estimating the parameters of this model, i.e., estimating L, α_l and τ_l for $l = 1, \ldots, L$. We refer to this approach as parametric channel estimation. We note that in this paper the specular channel assumption is used only for deriving our algorithm. The numerical investigation is conducted

using standardized and widely recognized channel models [12, 13]. It shows that parametric channel estimation works well even for diffuse channels, i.e., when the number of multipath components is large, such that each individual component cannot be resolved. As detailed in Sec. A.5, our approach performs well in this case by estimating a parsimonious representation of the channel frequency response (CFR) based on the model (A.1) which is valid within the system bandwidth.

Early works on parametric channel estimation address applications to underwater communications [14] and ultra-wideband (UWB) communications [15, 16]. Another classical example is the rake receiver [17]. All of these older works assume that the number of multipath components L is known a priori or use heuristics to estimate it.

A sparsity-based (or compressed sensing-based) approach can be used to allow for inherent estimation of the number of multipath components. Most literature on sparsity-based channel estimation [18–24] employs a grid-based approximation of the CIR model (A.1), where the multipath delays are confined to a discrete set of possible values. When a baud-spaced grid[1] is used, we refer to the samples of the CIR (A.1) as channel taps. The grid-based approximation results in a leakage effect [21, 25] and the vector of channel taps is therefore only approximately sparse [11, 23, 24]. We demonstrate in our numerical investigation that the grid-based approximation impairs the performance of receivers for OFDM systems with large modulation order operating in the high-SNR regime. From a compressed sensing point of view the effect of the grid-based approximation can be understood as a basis mismatch [26].

Recent works on off-grid compressed sensing have proposed methods that could in principle be applied to sparsity-based channel estimation without resorting to the grid approximation. These are based on atomic-norm minimization [27–29], finite rate of innovation [30] or Bayesian inference [31–34]. While all these methods show good performance, the former two cannot easily be incorporated in a turbo receiver. In this paper we show how sparsity-based parametric channel estimation can be incorporated in a turbo receiver by using approximate Bayesian inference. Our channel estimation scheme is *sparsity-based* in the sense that a sparsity-promoting prior model is used to achieve inherent estimation of the number of multipath components (the vector $\boldsymbol{z}$ associated with (A.8) is sparse) and it is *parametric* in the sense that a parametric channel model is used to design the channel estimator.

[1] In the baud-spaced grid, the distance between adjacent grid points is the reciprocal of the system bandwidth.

A.1.3 Prior Art

Several prior works incorporate sparsity-based channel estimation in a turbo receiver. Prasad et. al. [23, 24] propose a joint sparse channel estimation and *detection* scheme for OFDM transmission. Channel decoding is not considered in the joint processing and the EM algorithm is used for channel inference. A baud-spaced grid is used.

Turbo receiver design for OFDM systems via GAMP and relaxed BP is proposed by Schniter in [10, 11]. The estimated multipath delays are restricted to the baud-spaced grid. In the numerical evaluation of [10] the CIRs fulfill this restriction, thus avoiding the leakage effect at the expense of introducing an unrealistic channel model. In [11] a channel model generating continuous-valued delays is assumed. It is shown that the channel taps follow a super-Gaussian density which is modelled via a two-component Gaussian mixture. Due to the baud-spaced grid the channel taps are correlated, which is mimicked with a hidden Markov model. The resulting model has a large number of parameters to be estimated, which cause systems with high-order odulation format to exhibit a bit error rate (BER) floor when operating in the high-SNR regime (see Sec A.5).

The problem of parametric channel estimation based only on pilots or in the contrived case when the data symbols are given is equivalent to that of line spectral estimation [22]. The work [32] proposes a variational Bayesian approach to line spectral estimation. It is shown that the Bernoulli-Gaussian prior [35] is a powerful and tractable sparsity-inducing model. Our sparsity-based parametric channel estimator is inspired by [32] and uses the Bernoulli-Gaussian prior model too. It differs from [32] in several aspects: *a)* at the data subcarriers the observations are modulated with the unknown data symbols, *b)* we impose that the estimate of the posterior probability density function (pdf) of the multipath coefficients factorizes and *c)* to reduce computational complexity we use a point estimate of the multipath delays.[2]

A.1.4 Contributions

The contributions of this paper are as follows:

1. We propose a method to incorporate sparsity-based parametric channel estimation into a turbo receiver. Specifically we use the combined BP and MF (BP-MF) framework [8] to derive such a turbo receiver within a unified framework.

2. We show (numerically) that turbo-receivers for OFDM with high modulation order exhibit an error floor in the high-SNR regime when they

[2]By contrast, the scheme in [32] applied in our context computes estimates of the posterior distribution of the delays.

employ state-of-the-art sparse channel estimation based on the baud-spaced grid approximation. Our turbo receiver design demonstrates how this error floor can be avoided.

3. Our algorithm development demonstrates how the BP-MF framework can be modified to provide approximate ML estimation of model parameters and how some latent variables can be estimated jointly to improve convergence speed. We expect that these approaches will prove useful in other applications of BP-MF.

Our receiver only uses a few parameters (specifically the noise variance and the two parameters of the Bernoulli-Gaussian prior model, sparsity level ρ and multipath coefficient variance η) to describe the statistical properties of the wireless channel and these are inherently estimated by appropriately modifying BP-MF. This is in contrast to, for example, the linear MMSE (LMMSE) channel estimators, which require a-priori specification of the second-order statistics of the channel transfer function [4, 36], and the GAMP receiver [10], which relies on the second-order statistics of the channel taps and the transition probabilities of the hidden Markov model.

A.1.5 Notation and Outline

We denote column vectors as $\boldsymbol{a}$ and matrices as $\mathbf{A}$. Transposition is denoted as $(\cdot)^{\mathrm{T}}$ and conjugate (Hermitian) transposition as $(\cdot)^{\mathrm{H}}$. The scalar a_i or $[\boldsymbol{a}]_i$ gives the ith entry of vector $\boldsymbol{a}$, while $\boldsymbol{a}_{\mathcal{S}}$ gives a vector containing the entries in $\boldsymbol{a}$ at the indices in the integer set $\mathcal{S}$. The set difference operator $\mathcal{S}\backslash\{i\}$ gives the index set $\mathcal{S}$ with index i removed; we abuse notation slightly and write $\mathcal{S}\backslash i$ for short. The notation $[\mathbf{A}]_{i,k}$ gives the (i,k)th element of matrix $\mathbf{A}$. We denote the vector $\boldsymbol{a}$ with the ith element removed as $\boldsymbol{a}_{\backslash i}$ and use a similar notation for matrices with columns and/or rows removed (e.g. $[\mathbf{A}]_{i,\backslash k}$ for the ith row with kth entry removed). The notation $\mathrm{diag}(\boldsymbol{a})$ denotes a matrix with the entries of $\boldsymbol{a}$ on the diagonal and zeros elsewhere. The indicator function $\mathbb{1}_{[\cdot]}$ gives 1 when the condition in the brackets is fulfilled and 0 otherwise. The notation $a \propto^{\mathrm{e}} b$ denotes $\exp(a) \propto \exp(b)$, which implies $a = b + \text{const}$. The multivariate complex normal probability density function (pdf) is defined as

$$\mathrm{CN}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \triangleq \pi^{-\dim(\boldsymbol{x})}|\boldsymbol{\Sigma}|^{-1}\exp\Big(-(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$$

The notation $\mathrm{unif}(x;0,T)$ gives the continuous uniform pdf on the interval $[0,T]$ and $\mathrm{Bern}(x;\rho)$ gives the Bernoulli probability mass function (pmf) for $x \in \{0,1\}$ with probability of success ρ. We use $*$ to denote convolution and $\delta(\cdot)$ and $\delta[\cdot]$ to denote the Dirac and Kronecker delta, respectively.

The paper is structured as follows: In Section A.2 we specify the observation model. In Section A.3 our approach to approximate Bayesian inference is

discussed. The inference algorithm is derived in detail in Section A.4. Section A.5 presents the numerical evaluation. Conclusions are given in Section A.6.

A.2 Modelling

We consider data transmission using a single-input single-output OFDM system. Since we do not exploit any structure between consecutive OFDM symbols, we model the sequence of transmitted OFDM symbols to be independent and identically distributed (i.i.d.). The OFDM system transmits P pilot subcarriers and D data subcarriers, such that the total number of subcarriers per symbol is $N = P + D$. The sets $\mathcal{P}$ and $\mathcal{D}$ give the indices of the pilot and data subcarriers, respectively. It follows that $\mathcal{D} \cup \mathcal{P} = \{1, \ldots, N\}$ and $\mathcal{D} \cap \mathcal{P} = \emptyset$.

A.2.1 OFDM System

The K (equi-probable) information bits to be transmitted are stacked in vector $\boldsymbol{u} \in \{0,1\}^K$. These bits are coded by a rate R encoder and interleaved to get the length-K/R vector $\boldsymbol{c} = \mathcal{C}(\boldsymbol{u})$. The interleaving and coding function $\mathcal{C} : \{0,1\}^K \to \{0,1\}^{K/R}$ can represent any interleaver and coder, e.g. a turbo [37], low-density parity check (LDPC) [38] or convolutional code. We split $\boldsymbol{c}$ into subvectors $\boldsymbol{c}^{(i)} \in \{0,1\}^Q$, $i \in \mathcal{D}$, such that $\boldsymbol{c}^{(i)}$ contains the Q bits that are mapped to the ith subcarrier. The complex symbols $x_i = \mathcal{M}(\boldsymbol{c}^{(i)})$, $i \in \mathcal{D}$, are obtained via the 2^Q-ary mapping $\mathcal{M} : \{0,1\}^Q \to \mathbb{A}_\mathrm{D} \subset \mathbb{C}$, where $\mathbb{A}_\mathrm{D}$ is the data symbol alphabet. The pilots are selected in the pilot symbol alphabet $\mathbb{A}_\mathrm{P} \subset \mathbb{C}$. In OFDM, $\mathbb{A}_\mathrm{D}$ is typically a 2^Q-ary quadrature amplitude modulation (QAM) alphabet and $\mathbb{A}_\mathrm{P}$ a quadrature phase shift keying (QPSK) alphabet. The pilot and data symbols are stacked in vector $\boldsymbol{x}$. Vector $\boldsymbol{x}_\mathcal{D}$ contains the data symbols and $\boldsymbol{x}_\mathcal{P}$ contains the pilot symbols. We make the usual assumption in OFDM of time-limited CIR:

$$h(\tau) = 0 \quad \text{for} \quad \tau \notin [0; T_\mathrm{CP}], \tag{A.2}$$

where T_CP is the cyclic prefix duration.

By the assumption in (A.2) the OFDM system operates without inter-symbol interference, so we can consider a single OFDM symbol. The OFDM transmitter emits the baseband signal

$$s(t) = \begin{cases} \sum_{n=1}^{N} x_n \exp(j2\pi\Delta_f n t) & t \in [-T_\mathrm{CP}; T_\mathrm{sym}], \\ 0 & \text{otherwise}, \end{cases} \tag{A.3}$$

where Δ_f gives the subcarrier spacing and $T_\mathrm{sym} = \Delta_f^{-1}$ is the OFDM symbol

length. The signal at the output of receiver RF front-end reads

$$r(t) = h(\tau) * s(t) + \tilde{w}(t), \tag{A.4}$$

where $\tilde{w}(t)$ is low-pass filtered white Gaussian noise. The receiver then samples $r(t)$, removes the cyclic prefix and calculates the discrete Fourier transform to obtain the observed vector $\boldsymbol{y}$. The assumption in (A.2) ensures that orthogonality of the subcarriers is preserved. It can be shown [39] that

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{h} + \boldsymbol{w}, \tag{A.5}$$

where $\mathbf{X} = \mathrm{diag}(\boldsymbol{x})$. The Gaussian noise vector $\boldsymbol{w}$ is assumed[3] white with component variance β. The vector $\boldsymbol{h}$ contains the CFR at the subcarrier frequencies. Its entries are samples of the (continuous) Fourier transform of $h(\tau)$:

$$h_n = \int_0^{T_{\mathrm{sym}}} h(\tau) \exp(-j2\pi\Delta_f n\tau)\, \mathrm{d}\tau, \quad n = 1, \ldots, N. \tag{A.6}$$

Now, rewrite (A.1) as a convolution, insert it into (A.6) and use the convolution theorem to get

$$\boldsymbol{h} = \mathbf{C}\boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\alpha}, \tag{A.7}$$

where $\mathbf{C}$ is a diagonal matrix with its diagonal made of samples of the Fourier transform of $c(\tau)$ at the subcarrier frequencies (i.e., the diagonal entries have a form similar to (A.6)). The matrix $\boldsymbol{\Psi}(\boldsymbol{\tau}) \in \mathbb{C}^{N\times L}$ has (n,l)th entry $\exp(-j2\pi\Delta_f n\tau_l)$, $n = 1, \ldots, N$, $l = 1, \ldots, L$. For ease of notation, we define $\boldsymbol{\psi}(\tau_l)$ as the lth column of $\boldsymbol{\Psi}(\boldsymbol{\tau})$. We have stacked the channel multipath coefficients and delays into vectors $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_L]^{\mathrm{T}} \in \mathbb{C}^L$ and $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_L]^{\mathrm{T}} \in [0, T_{\mathrm{CP}}]^L$. In the following we make the common assumption that $\mathbf{C} = \mathbf{I}$, i.e., that the combined response of the transmit and receive filters is flat over the system bandwidth. If this is not the case, the effects of the filters can easily be included in the dictionary matrix $\boldsymbol{\Psi}(\boldsymbol{\tau})$, as shown in [40].

We recognize from (A.7) that $\boldsymbol{h}$ is a superposition of complex sinusoids. Thus, given the data symbols in $\mathbf{X}$, the estimation of L, $\boldsymbol{\alpha}$ and $\boldsymbol{\tau}$ reduces to an instance of line spectral estimation.

A.2.2 The Specular Channel Assumption

We say that the wireless channel modelled by (A.1) is specular when $L \ll \lceil T_{\mathrm{CP}} N \Delta_f \rceil$ and roughly speaking the multipath components are well separated.[4] The criterion $L \ll \lceil T_{\mathrm{CP}} N \Delta_f \rceil$ indicates that the number of components is much smaller than the number of degrees of freedom [18] of the CFR

[3] This assumption is fulfilled when the filter in the receive RF front-end has constant frequency response within the system bandwidth.

[4] "Well separated" is here meant relative to the inverse of the system bandwidth $1/(N\Delta_f)$.

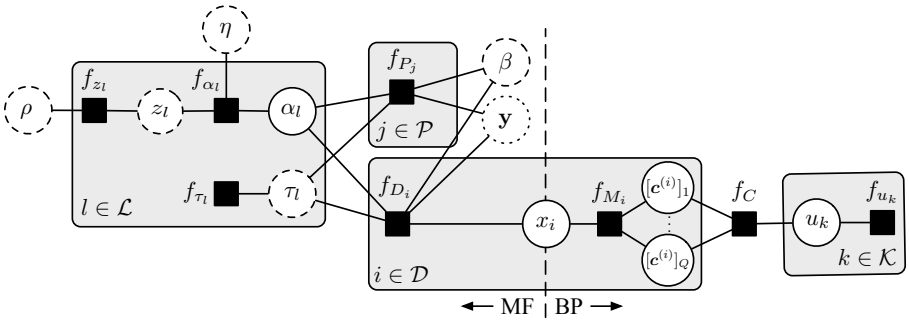

Fig. A.2: Factor graph representation of the probabilistic model describing the complete OFDM system and channel model. The shaded areas indicate multiple copies of the nodes, as specified by the index sets. The vector of observations $\boldsymbol{y}$ is included with a dotted line because it is known at the time of inference. Variables of which a point estimate is obtained (as opposed to a full variational estimate of the posterior pdf) are represented by circles with dashed line. The vertical dashed line shows the separation between the BP and MF subgraphs.

vector $\boldsymbol{h}$ under the assumption in (A.2). On the other hand, when the number of components is much larger than the number of degrees of freedom ($L \gg \lceil T_{\text{CP}} N \Delta_f \rceil$) or when they are located closely together, we refer to the channel as diffuse.

Empirical evidence suggests that the wireless channel in some propagation environments is specular to a large degree. In practice, specular channels are composed of a small number of dominant multipath components and a remaining part with power below the noise floor. Examples include the ultra-wideband channels that are considered for 5G wireless communications [41, 42] and underwater acoustic channels [43]. See also [18, 44] and references therein.

In this paper we derive a receiver based on the specular channel assumption. This may seem like a very restricting assumption. However, as demonstrated in Sec. A.5, our algorithm performs very well with diffuse channels. In particular we perform numerical experiments with a channel where the multipath components appear in clusters with many components in each cluster ($L \gg \lceil T_{\text{CP}} N \Delta_f \rceil$), such that the individual components in (A.1) can no longer be resolved. In this case the specular channel assumption forces the algorithm to find a parsimonious representation of the CFR within the system bandwidth.

A.2.3 Probabilistic Model of the OFDM System

We are now ready to present a probabilistic model which describes the complete OFDM system. The model expresses the joint probability of all variables in the system as a product of factors. This factorization of the joint probabil-

ity is represented as the factor graph depicted in Fig. A.2. In the following we introduce the variables and factors in the factor graph, moving from right to left.[5]

The interleaving, coding and modulation of the data bits are described in Sec. A.2.1. The subgraph characterizing the system implementing these tasks involves the factors

$$\begin{aligned} f_{u_k}(u_k) &\triangleq p(u_k) = 0.5\,\mathbb{1}_{[u_k\in\{0,1\}]}, && k \in \mathcal{K}, \\ f_C(\boldsymbol{c},\boldsymbol{u}) &\triangleq p(\boldsymbol{c}|\boldsymbol{u}) = \mathbb{1}_{[\boldsymbol{c}=\mathcal{C}(\boldsymbol{u})]}, \\ f_{M_i}(x_i,\boldsymbol{c}^{(i)}) &\triangleq p(x_i|\boldsymbol{c}^{(i)}) = \mathbb{1}_{[x_i=\mathcal{M}(\boldsymbol{c}^{(i)})]}, && i \in \mathcal{D}, \end{aligned}$$

where $\mathcal{K} = \{1,\dots,K\}$ is the index set of the information bits. The factor $f_C(\boldsymbol{c},\boldsymbol{u})$ describes the interleaving and channel coding processes. By "zooming in" this factor can be expanded to a subgraph involving auxiliary variables and factors which describe the structure of the channel code and interleaver.

The subgraph characterizing the observation process described by (A.5) and (A.7) involves the following factors for pilot- and data subcarriers, respectively:

$$\begin{aligned} f_{P_j}(\boldsymbol{\alpha},\boldsymbol{\tau},\beta) &\triangleq p(y_j|\boldsymbol{\alpha},\boldsymbol{\tau};\beta) \\ &= \mathrm{CN}(y_j; x_j[\boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\alpha}]_j,\beta), && j \in \mathcal{P}, \\ f_{D_i}(x_i,\boldsymbol{\alpha},\boldsymbol{\tau},\beta) &\triangleq p(y_i|x_i,\boldsymbol{\alpha},\boldsymbol{\tau};\beta) \\ &= \mathrm{CN}(y_i; x_i[\boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\alpha}]_i,\beta), && i \in \mathcal{D}. \end{aligned}$$

To model the specular channel assumption (i.e., that (A.1) is composed of a small number of multipath components), we use a Bernoulli-Gaussian prior which assigns large probability to the event $\{\alpha_l = 0\}$. We model $L_{\max}$ multipath components of which only a subset is activated, i.e. has $\alpha_l \neq 0$. The number $L_{\max}$ is an upper bound on the number of multipath components that can be estimated.[6] This allows us to derive an algorithm which inherently estimates the number of multipath components. Each component is assigned an activation variable $z_l \in \{0,1\}$, which is 1 when said multipath component is active and 0 otherwise. The sequence $\{z_1,\dots,z_{L_{\max}}\}$ is modelled i.i.d. where each z_l is assigned a Bernoulli prior with activation probability ρ:

$$f_{z_l}(z_l,\rho) \triangleq p(z_l;\rho) = \mathrm{Bern}(z_l;\rho), \qquad l \in \mathcal{L}, \tag{A.8}$$

[5] We abuse terminology and associate variables and factors with their respective nodes in the factor graph.

[6] In our implementation we select $L_{\max} = \lceil T_{\mathrm{CP}} N \Delta_f \rceil + 1$, which is the maximum number of degrees of freedom under the assumption (A.2) [18]. It roughly corresponds to the number of baud-spaced (spacing $1/(N\Delta_f)$) components on the interval $[0, T_{\mathrm{CP}})$.

where we have defined the set of multipath component indices $\mathcal{L} = \{1, \ldots, L_{\max}\}$. The prior density of the multipath coefficient α_l is conditioned on z_l, such that $z_l = 0$ implies $\alpha_l = 0$ and $z_l = 1$ gives a Gaussian density with variance η:

$$\begin{aligned} f_{\alpha_l}(\alpha_l, z_l, \eta) &\triangleq p(\alpha_l | z_l; \eta) \\ &= (1 - z_l)\delta(\alpha_l) + z_l \, \mathrm{CN}(\alpha_l; 0, \eta), \qquad l \in \mathcal{L}. \end{aligned}$$

When performing inference in this model, the estimated number of active multipath components is $\hat{L} \triangleq ||\hat{\boldsymbol{\alpha}}||_0$, where $\hat{\boldsymbol{\alpha}}$ is a vector containing the estimates of α_l for all $l \in \mathcal{L}$.

We finally need to impose a prior model on the multipath delays τ_l, $l \in \mathcal{L}$. The only prior information available is through the assumption that for all $l \in \mathcal{L}$ we have $0 \leq \tau_l \leq T_{\mathrm{CP}}$, so an i.i.d. uniform prior is used:

$$f_{\tau_l}(\tau_l) \triangleq p(\tau_l) = \mathrm{unif}(\tau_l; 0, T_{\mathrm{CP}}), \qquad l \in \mathcal{L}.$$

A.3 Inference Method

The BER optimal receiver (assuming ρ, η and β known) computes the MAP estimate

$$\hat{u}_k = \underset{u_k \in \{0,1\}}{\arg\max}\, p(u_k | \boldsymbol{y}; \rho, \eta, \beta), \quad k \in \mathcal{K}. \tag{A.9}$$

The pdf $p(u_k | \boldsymbol{y}; \rho, \eta, \beta) \propto p(u_k, \boldsymbol{y}; \rho, \eta, \beta)$ can ideally be found by marginalizing all variables but u_k in the joint pdf

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{x}_{\mathcal{D}}, \boldsymbol{c}, \boldsymbol{u}; \rho, \eta, \beta) &= p(\boldsymbol{y} | \boldsymbol{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \boldsymbol{\tau}; \beta) \\ &\prod_{l \in \mathcal{L}} p(\alpha_l | z_l; \eta) p(z_l; \rho) p(\tau_l) \prod_{i \in \mathcal{D}} p(x_i | \boldsymbol{c}^{(i)}) p(\boldsymbol{c} | \boldsymbol{u}) \prod_{k \in \mathcal{K}} p(u_k). \end{aligned}$$

Calculating the marginals of u_k, $k \in \mathcal{K}$, is intractable and we resort to approximate Bayesian inference.

A.3.1 Combined Belief Propagation and Mean-Field

Our inference method is based on the merged belief propagation and mean-field (BP-MF) framework of [8]. In this framework a so-called belief function is found for each variable in the factor graph. The belief function is an approximation of the marginal posterior pdf or pmf of that variable. We abuse notation and let $q(a)$ denote the belief of variable a. When the set of belief functions has been calculated, the MAP estimate of the kth data bit is found as the mode of $q(u_k)$.

For tractability we obtain a point estimate of the variables $\boldsymbol{z}$ and $\boldsymbol{\tau}$. This is achieved as proposed in [8] by restricting their beliefs to be Kronecker and Dirac delta functions, i.e., $q(z_l) = \delta[z_l - \hat{z}_l]$ and $q(\tau_l) = \delta(\tau_l - \hat{\tau}_l)$ for all $l \in \mathcal{L}$.

At the heart of BP-MF lies the so-called region-based free energy approximation (RBFE) [45]. The RBFE is obtained by splitting the factor graph into a MF and a BP subgraph, as indicated in Fig. A.2. The RBFE is a function of[7] the point estimates $\hat{\boldsymbol{z}}, \hat{\boldsymbol{\tau}}$ and the belief functions $q(\alpha_l), q(x_i), q([\boldsymbol{c}^{(i)}]_m)$ and $q(u_k)$ for indices $l \in \mathcal{L}, i \in \mathcal{D}, k \in \mathcal{K}$ and $m = 1, \ldots, Q$. It is also a function of the model parameter estimates $(\hat{\rho}, \hat{\eta}, \hat{\beta})$, as justified below. The expression of the RBFE is given in Appendix A. BP-MF seeks to minimize the RBFE under a number of normalization and consistency constraints. The messages of BP-MF are derived such that at convergence they satisfy the Karush-Kuhn-Tucker conditions of the constrained RBFE minimization, i.e., a (possibly local) minimum of the constrained problem is found. See [8] for a more detailed discussion of BP-MF.

The understanding of BP-MF as RBFE minimization allows us to make a number of adaptations to the message-passing scheme to improve convergence speed. Further, we will see that this understanding is useful when analyzing convergence of the algorithm.

A.3.2 Model Parameter Estimation with BP-MF

The BP-MF framework [8] does not directly provide a method to estimate the unknown model parameters (ρ, η, β). We propose to do so by letting the RBFE be a function of these model parameters. The model parameter estimates $(\hat{\rho}, \hat{\eta}, \hat{\beta})$ are then obtained as the minimizers of the RBFE.

To justify this method we first note that the model parameters are located in the MF subgraph. Then we follow an approach similar to [46] to obtain a lower bound on the marginal log-likelihood function:

$$\ln p(\boldsymbol{y}; \hat{\rho}, \hat{\eta}, \hat{\beta}) \geq -F_{\text{BP-MF}} + \text{const.}, \tag{A.10}$$

where $F_{\text{BP-MF}}$ is the RBFE (A.35) and the constant only depends on beliefs of variables in the BP subgraph (including $q(x_i)$, for $i \in \mathcal{D}$), i.e., it does not depend on $(\hat{\rho}, \hat{\eta}, \hat{\beta})$. It can then be seen that the values of $(\hat{\rho}, \hat{\eta}, \hat{\beta})$ minimizing $F_{\text{BP-MF}}$ maximize the lower bound on the likelihood function in (A.10). These minimizers are thus approximate ML estimates. We note that if the above approach is applied in a pure MF context it simplifies to variational EM estimation with all other variables treated as latent variables [8, 47].

[7] The RBFE is also a functional of the beliefs corresponding to the factors in the BP subgraph. BP-MF enforces consistency between the variable beliefs and these factor beliefs. Since the latter are not relevant to the derivation of the receiver, we omit them.

A.3.3 Relation to Prior Art

To relate our receiver algorithm to current methods we note that the decoding of many popular channel codes can be described as an instance of BP [48] in a factor graph [49–51]. For example, BP decoding of a convolutional code leads to the BCJR algorithm [52]. We see in Fig. A.2 that the merged BP-MF algorithm employs BP in the subgraph which represents the channel code, i.e., standard techniques are used for decoding.

Similarly, there are examples in the literature of MF inference where the underlying factor graph resembles the MF subgraph of our receiver. The work [32] uses a Bernoulli-Gaussian prior model similar to that in our work, while [31], [33] use a gamma-Gaussian prior typical of sparse Bayesian learning.

The strength of the BP-MF framework is now clear: It allows us to merge existing methods for channel decoding and sparsity-based estimation into one unified receiver algorithm, which formally seeks to minimize the RBFE.

A.4 Parametric BP-MF Receiver

To minimize the RBFE, we apply the BP-MF algorithm given by Eq. (21)–(22) in [8] on the factor graph of Fig. A.2. In the following we use the notation $\langle \cdot \rangle_a$ to denote expectation with respect to the belief density $q(a)$. We follow the convention of [8] in naming the messages. In [9] a similar BP-MF receiver is derived, which does not exploit channel sparsity.

A.4.1 Message Passing for Channel Estimation

Update of Coefficient Belief

We start by finding belief updates in the MF subgraph. To find the update of $q(\alpha_l)$, $l \in \mathcal{L}$, we calculate the messages passed to the node α_l:

$$
\begin{aligned}
m^{\mathrm{MF}}_{f_{\alpha_l} \to \alpha_l}(\alpha_l) &\propto \begin{cases} \exp\left(-\hat{\eta}^{-1}|\alpha_l|^2\right) & \text{if } \hat{z}_l = 1, \\ \delta(\alpha_l) & \text{if } \hat{z}_l = 0 \end{cases} \\
m^{\mathrm{MF}}_{f_{D_i} \to \alpha_l}(\alpha_l) &\propto \exp\left(-\hat{\beta}^{-1} \left\langle |y_i - x_i [\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\boldsymbol{\alpha}]_i|^2 \right\rangle_{x_i, \boldsymbol{\alpha}_{\setminus l}} \right) \\
m^{\mathrm{MF}}_{f_{P_j} \to \alpha_l}(\alpha_l) &\propto \exp\left(-\hat{\beta}^{-1} \left\langle |y_j - x_j [\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\boldsymbol{\alpha}]_j|^2 \right\rangle_{\boldsymbol{\alpha}_{\setminus l}} \right),
\end{aligned}
$$

which holds for all $l \in \mathcal{L}$, $i \in \mathcal{D}$ and $j \in \mathcal{P}$. Taking the product of all messages going into the node α_l gives its belief

$$q(\alpha_l) = \begin{cases} \mathrm{CN}(\alpha_l; \hat{\mu}_l, \hat{\sigma}_l^2) & \text{if } \hat{z}_l = 1, \\ \delta(\alpha_l) & \text{if } \hat{z}_l = 0, \end{cases} \tag{A.11}$$

with the active component mean and variance

$$\hat{\mu}_l = \hat{\sigma}_l^2 q_l \tag{A.12}$$

$$\hat{\sigma}_l^2 = \left(s_l + \hat{\eta}^{-1}\right)^{-1}. \tag{A.13}$$

Here we have introduced

$$s_l = \hat{\beta}^{-1} \boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l) \left\langle \mathbf{X}^{\mathrm{H}} \mathbf{X} \right\rangle_{\boldsymbol{x}_\mathcal{D}} \boldsymbol{\psi}(\hat{\tau}_l) \tag{A.14}$$

$$q_l = \hat{\beta}^{-1} \boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l) \boldsymbol{r} \tag{A.15}$$

$$\boldsymbol{r} = \langle \mathbf{X} \rangle_{\boldsymbol{x}_\mathcal{D}}^{\mathrm{H}} \boldsymbol{y} - \left\langle \mathbf{X}^{\mathrm{H}} \mathbf{X} \right\rangle_{\boldsymbol{x}_\mathcal{D}} \boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}} \setminus l}) \hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}} \setminus l}. \tag{A.16}$$

Note that the belief of inactive components ($\hat{z}_l = 0$) becomes a point mass at $\alpha_l = 0$, thus eliminating the influence of that component in the product $\mathbf{X}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\boldsymbol{\alpha}$. We have defined the set of currently active components as $\hat{\mathcal{A}} \triangleq \{l : \hat{z}_l = 1\}$ and the vectors $\hat{\boldsymbol{\mu}} = [\hat{\mu}_1, \ldots, \hat{\mu}_{L_{\max}}]^{\mathrm{T}}$, $\hat{\boldsymbol{\sigma}}^2 = [\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{L_{\max}}^2]^{\mathrm{T}}$.

Joint Update of Delay and Coefficient Belief

We now turn our attention to the estimation of the multipath delays τ_l, $l \in \mathcal{L}$. To improve the convergence speed of the algorithm, we find the update of $\hat{\tau}_l$ by minimizing the RBFE *jointly* with respect to the beliefs $q(\alpha_l)$ and $\hat{\tau}_l$. Due to the selected prior $p(\tau_l)$, the following expressions are valid for $\hat{\tau}_l \in [0, T_{\mathrm{CP}}]$. We are only concerned with active components, i.e., $l \in \hat{\mathcal{A}}$ and thus $\hat{z}_l = 1$. Writing only the terms of the RBFE (A.35) which depend on $q(\alpha_l)$ and $\hat{\tau}_l$, we get

$$F_{\text{BP-MF}}(q(\alpha_l), \hat{\tau}_l) \propto^{\mathrm{e}} \int q(\alpha_l) \ln \frac{q(\alpha_l)}{Q(\alpha_l, \hat{\tau}_l)} \, \mathrm{d}\alpha_l \tag{A.17}$$

with

$$\begin{aligned} Q(\alpha_l, \hat{\tau}_l) &= p(\alpha_l | \hat{z}_l; \hat{\eta}) p(\hat{\tau}_l) \exp\left(\left\langle \ln p(\boldsymbol{y} | \boldsymbol{x}_\mathcal{D}, \boldsymbol{\alpha}, \hat{\boldsymbol{\tau}}; \hat{\beta}) \right\rangle_{\boldsymbol{x}_\mathcal{D}, \boldsymbol{\alpha}_{\setminus l}} \right) \\ &\propto \mathrm{CN}(\alpha_l; \hat{\mu}_l, \hat{\sigma}_l^2) \exp\left(\frac{|q_l|^2}{s_l + \hat{\eta}^{-1}} \right), \end{aligned} \tag{A.18}$$

where $\hat{\sigma}_l^2$, $\hat{\mu}_l$, s_l and q_l are given by (A.12) - (A.15) and thus implicitly are functions of $\hat{\tau}_l$. We need to minimize (A.17) under the normalization constraint $\int q(\alpha_l)\,\mathrm{d}\alpha_l = 1$. To do so, define

$$g_{\tau_l}(\hat{\tau}_l) \triangleq \max_{\tilde{q}(\alpha_l):\int \tilde{q}(\alpha_l)\,\mathrm{d}\alpha_l = 1} -F_{\text{BP-MF}}\left(\tilde{q}(\alpha_l), \hat{\tau}_l\right) \tag{A.19}$$

$$\propto^{\mathrm{e}} \ln \int Q(\alpha_l, \hat{\tau}_l)\,\mathrm{d}\alpha_l \tag{A.20}$$

$$\propto^{\mathrm{e}} \frac{\hat{\beta}^{-2}}{s_l + \hat{\eta}^{-1}} |\boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l)\boldsymbol{r}|^2. \tag{A.21}$$

The result in (A.20) is easily obtained by noting that (A.17) can be rewritten as

$$F_{\text{BP-MF}} \propto^{\mathrm{e}} \mathrm{KL}\left[q(\alpha_l)\middle|\middle|\frac{Q(\alpha_l, \hat{\tau}_l)}{\int Q(\tilde{\alpha}_l, \hat{\tau}_l)\,\mathrm{d}\tilde{\alpha}_l}\right] - \ln \int Q(\tilde{\alpha}_l, \hat{\tau}_l)\,\mathrm{d}\tilde{\alpha}_l,$$

where $\mathrm{KL}[\cdot||\cdot]$ is the Kullback-Leibler divergence. The coefficient belief is selected as the maximizer of (A.19), i.e., $q(\alpha_l) = Q(\alpha_l, \hat{\tau}_l) / \int Q(\tilde{\alpha}_l, \hat{\tau}_l)\,\mathrm{d}\tilde{\alpha}_l$, which is easily shown to coincide with the result in (A.11).

Since s_l is constant with respect to $\hat{\tau}_l$, we find the delay update as

$$\hat{\tau}_l = \arg\max_{\tilde{\tau}_l \in [0, T_{\text{CP}}]} g_{\tau_l}(\tilde{\tau}_l) = \arg\max_{\tilde{\tau}_l \in [0, T_{\text{CP}}]} |\boldsymbol{\psi}^{\mathrm{H}}(\tilde{\tau}_l)\boldsymbol{r}|^2. \tag{A.22}$$

We recognize the objective function in (A.22) as the continuous periodogram of the residual vector $\boldsymbol{r}$. While it is possible to find the maximizer of the periodogram, doing so has high computational cost. In our iterative algorithm, we instead find an update of $\hat{\tau}_l$ which cannot increase the objective in (A.22). Denote the updated delay estimate as $\hat{\tau}_l^{[t]}$ and the previous delay estimate as $\hat{\tau}_l^{[t-1]}$. Our scheme now reads:

1. Find initial step $\Delta = \frac{g'_\tau(\hat{\tau}_l^{[t-1]})}{|g''_\tau(\hat{\tau}_l^{[t-1]})|}$.
2. If $g_\tau(\hat{\tau}_l^{[t-1]} + \Delta) \geq g_\tau(\hat{\tau}_l^{[t-1]})$, set $\hat{\tau}_l^{[t]} = \hat{\tau}_l^{[t-1]} + \Delta$ and terminate. Otherwise set $\Delta = \frac{\Delta}{2}$ and repeat step 2.

Functions $g'_\tau(\tau_l)$ and $g''_\tau(\tau_l)$ are the first and second derivatives of (A.21). The scheme gives the Newton update of $\hat{\tau}_l$ if this value increases the objective function and otherwise resorts to a gradient ascent with a backtracking line search. We have the following lemma, which we will use in the convergence analysis:

Lemma A.1. *The procedure listed in Steps 1-2 above followed by an update of $q(\alpha_l)$ does not increasing the RBFE.*

Proof: First, note that the updated $\hat{\tau}_l$, does not decrease $g_{\tau_l}(\hat{\tau}_l)$. It then follows that by selecting the maximizer of (A.19), the RBFE is non-increasing. ■

Joint Update of Activation Variable and Coefficient Belief

We now turn our focus on the update of the activation variable $\hat{z}_l$. It is again desirable to perform a joint update of $\hat{z}_l$ and $q(\alpha_l)$. We proceed in a similar way as we did to compute the updates of the multipath delays. The terms in the RBFE (A.35) which depend on $q(\alpha_l)$ and $\hat{z}_l$ are denoted as $F_{\text{BP-MF}}(q(\alpha_l), \hat{z}_l)$. We then define

$$g_{z_l}(\hat{z}_l) \triangleq \max_{\tilde{q}(\alpha_l):\int \tilde{q}(\alpha_l)\,\mathrm{d}\alpha_l=1} -F_{\text{BP-MF}}(\tilde{q}(\alpha_l), \hat{z}_l) \tag{A.23}$$

$$\propto^{\mathrm{e}} \begin{cases} \frac{|\hat{\mu}_l|^2}{\hat{\sigma}_l^2} + \ln\frac{\hat{\sigma}_l^2}{\hat{\eta}} + \ln\hat{\rho} & \text{if } \hat{z}_l = 1, \\ \ln(1-\hat{\rho}) & \text{if } \hat{z}_l = 0. \end{cases} \tag{A.24}$$

This result is easily obtained by following steps analogous to (A.17) – (A.21). The activation variable solves the decision problem $\hat{z}_l = \max_{\tilde{z}_l\in\{0,1\}} g_{z_l}(\tilde{z}_l)$. Writing the "activation criterion" $g_{z_l}(1) > g_{z_l}(0)$ we get

$$\frac{|\hat{\mu}_l|^2}{\hat{\sigma}_l^2} > \ln\frac{\hat{\eta}}{\hat{\sigma}_l^2} + \ln\frac{1-\hat{\rho}}{\hat{\rho}}. \tag{A.25}$$

If the above criterion is true we set $\hat{z}_l = 1$; otherwise we set $\hat{z}_l = 0$. The corresponding update of $q(\alpha_l)$ is the maximizer of (A.23), which remains as in (A.11). The criterion in (A.25) is the same as that obtained in [32].

Update of Channel Parameter Estimates

The channel parameters (ρ, η, β) are estimated as the values which minimize the RBFE. Writing only the terms of the RBFE (A.35) which depend on the channel parameters we have

$$\begin{aligned} F_{\text{BP-MF}}(\hat{\rho}, \hat{\eta}, \hat{\beta}) &\propto^{\mathrm{e}} -\left\langle \ln\prod_{l\in\mathcal{L}} p(\hat{z}_l;\hat{\rho})p(\alpha_l|\hat{z}_l;\hat{\eta})p(\boldsymbol{y}|\boldsymbol{\alpha},\hat{\boldsymbol{\tau}},\boldsymbol{x}_{\mathcal{D}};\hat{\beta})\right\rangle_{\boldsymbol{x}_{\mathcal{D}},\boldsymbol{\alpha}} \\ &\propto^{\mathrm{e}} ||\hat{\boldsymbol{z}}||_0 \ln\hat{\rho} + (L_{\max} - ||\hat{\boldsymbol{z}}||_0)\ln(1-\hat{\rho}) - N\ln\hat{\beta} - \hat{\beta}^{-1}u \\ &\qquad - ||\hat{\boldsymbol{z}}||_0 \ln\hat{\eta} - \hat{\eta}^{-1}\sum_{l:\hat{z}_l=1}(|\hat{\mu}_l|^2 + \hat{\sigma}_l^2), \end{aligned} \tag{A.26}$$

where

$$\begin{aligned} u &\triangleq \left\langle ||\boldsymbol{y} - \mathbf{X}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\boldsymbol{\alpha}||^2\right\rangle_{\boldsymbol{x}_{\mathcal{D}},\boldsymbol{\alpha}} \\ &= ||\boldsymbol{y}||^2 + \hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}^{\mathrm{H}}\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\boldsymbol{x}_{\mathcal{D}}}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}} + \sum_{l\in\hat{\mathcal{A}}}\hat{\sigma}_l^2\boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l)\left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\boldsymbol{x}_{\mathcal{D}}}\boldsymbol{\psi}(\hat{\tau}_l) \end{aligned}$$

$$-2\,\mathrm{Re}\Big\{\boldsymbol{y}^{\mathrm{H}}\,\langle\mathbf{X}\rangle_{x_{\mathcal{D}}}\,\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}\Big\}. \quad \text{(A.27)}$$

It is readily seen that $F_{\text{BP-MF}}(\hat{\rho},\hat{\eta},\hat{\beta})$ can be minimized independently with respect to each of the parameters. By taking derivatives and equating to zero we find the global minima (the second derivatives are all positive):

$$\hat{\rho} = \frac{||\hat{\boldsymbol{z}}||_0}{L_{\max}} \quad \text{(A.28)}$$

$$\hat{\eta} = \frac{\sum_{l:\hat{z}_l=1}(|\hat{\mu}_l|^2+\hat{\sigma}_l^2)}{||\hat{\boldsymbol{z}}||_0} \quad \text{(A.29)}$$

$$\hat{\beta} = \frac{u}{N}. \quad \text{(A.30)}$$

Iterating all Coefficient Beliefs Ad-Infinitum

In [33] it is demonstrated that iterating the updates of some variables *ad-infinitum* is a powerful technique for increasing the convergence speed of MF algorithms. We apply that idea to the beliefs of the multipath coefficients.

Since $q(\alpha_l) = \delta(\alpha_l)$ for all $l \in \mathcal{L}\backslash\hat{\mathcal{A}}$, the following discussion is only concerned with the beliefs of active components, i.e. for $l \in \hat{\mathcal{A}}$. First note that the variance (A.13) of an active multipath coefficient $\hat{\sigma}_l^2$ does not depend on the beliefs of the remaining coefficients $q(\alpha_k)$, $k \neq l$. The mean (A.12) of the lth coefficient, on the other hand, depends on the remaining mean values as

$$\hat{\mu}_l = \underbrace{\hat{\sigma}_l^2}_{[\mathbf{Q}]_{l,l}^{-1}}\left(\underbrace{\hat{\beta}^{-1}\boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l)\langle\mathbf{X}\rangle_{x_{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{y}}_{p_l} - \sum_{k\in\hat{\mathcal{A}}\backslash l}\underbrace{\hat{\beta}^{-1}\boldsymbol{\psi}^{\mathrm{H}}(\hat{\tau}_l)\langle\mathbf{X}^{\mathrm{H}}\mathbf{X}\rangle_{x_{\mathcal{D}}}\boldsymbol{\psi}(\hat{\tau}_k)}_{[\mathbf{Q}]_{l,k}}\hat{\mu}_k\right),$$

for all $l \in \hat{\mathcal{A}}$. The matrix $\mathbf{Q}$ is of size $|\hat{\mathcal{A}}| \times |\hat{\mathcal{A}}|$ and we have abused notation in using l,k as indices into this matrix, because $1 \leq l,k \leq L_{\max}$, even though $|\hat{\mathcal{A}}| \leq L_{\max}$. The above equation is recognized as the Gauss-Seidel [53] iteration for solving the system of linear equations

$$\mathbf{Q}\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}} = \boldsymbol{p} \quad \text{(A.31)}$$

with

$$\boldsymbol{p} = \hat{\beta}^{-1}\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\,\langle\mathbf{X}\rangle_{x_{\mathcal{D}}}^{\mathrm{H}}\,\boldsymbol{y}$$
$$\mathbf{Q} = \hat{\beta}^{-1}\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\left\langle\mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{x_{\mathcal{D}}}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}}) + \hat{\eta}^{-1}\mathbf{I}.$$

It follows that the updates of $\hat{\mu}_l$, for all $l \in \hat{\mathcal{A}}$, converge to the solution $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}$ found by solving (A.31).

We note that in the hypothetical special case where the beliefs of $\mathbf{X}$ are point estimates (or equivalently known) $\boldsymbol{y} = \mathbf{X}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\boldsymbol{\alpha}_{\hat{\mathcal{A}}} + \boldsymbol{w}$ is a linear observation model with Gaussian noise. In this case, the estimator $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}} = \mathbf{Q}^{-1}\boldsymbol{p}$ reduces to the LMMSE estimator of $\boldsymbol{\alpha}_{\hat{\mathcal{A}}}$ in the linear observation model under the Bayesian model dictated by the current beliefs of the remaining variables. The estimator $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}} = \mathbf{Q}^{-1}\boldsymbol{p}$ is, however, not the LMMSE estimator of $\boldsymbol{\alpha}_{\hat{\mathcal{A}}}$ when the uncertainty of the estimate of $\mathbf{X}$ is considered.

A.4.2 Message-Passing for Decoding

In the previous subsections we derived the *belief functions* $q(\cdot)$ of the variables whose factor neighbours are in the MF subgraph only. To perform inference in the BP subgraph, i.e., detection, demapping, decoding and deinterleaving, we need to calculate the *messages* that are passed along its edges.

We begin with the messages $n_{x_i \to f_{M_i}}(x_i)$, $i \in \mathcal{D}$, which constitute the interface from the continuous-valued channel estimator to the discrete-valued decoder. They are given as

$$n_{x_i \to f_{M_i}}(x_i) = m^{\mathrm{MF}}_{f_{D_i} \to x_i}(x_i) \propto \mathrm{CN}\left(x_i; \frac{y_i \langle h_i \rangle^*_{\boldsymbol{\alpha},\boldsymbol{\tau}}}{\left\langle |h_i|^2 \right\rangle_{\boldsymbol{\alpha},\boldsymbol{\tau}}}, \frac{\hat{\beta}}{\left\langle |h_i|^2 \right\rangle_{\boldsymbol{\alpha},\boldsymbol{\tau}}}\right), \tag{A.32}$$

where $h_i \triangleq [\boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\alpha}]_i$ is the channel frequency response at subcarrier i. Its mean and second moment are

$$\begin{aligned} \langle h_i \rangle_{\boldsymbol{\alpha},\boldsymbol{\tau}} &= [\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\hat{\boldsymbol{\mu}}]_i \\ \left\langle |h_i|^2 \right\rangle_{\boldsymbol{\alpha},\boldsymbol{\tau}} &= \left[\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})(\hat{\boldsymbol{\mu}}\hat{\boldsymbol{\mu}}^{\mathrm{H}} + \mathrm{diag}(\hat{\sigma}^2))\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}})\right]_{i,i}. \end{aligned}$$

Note that even though the above expression has the form of a Gaussian, the messages are probability mass functions obtained by evaluating the above Gaussian at the points of the symbol alphabet $\mathbb{A}_{\mathrm{D}}$ followed by appropriate normalization.

The mean in (A.32) can be interpreted as the output of an LMMSE equalizer. Consider the observation model $y_i = h_i x_i + w_i$ where $p(w_i) = \mathrm{CN}(w_i; 0, \hat{\beta})$ and $h_i = [\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\boldsymbol{\alpha}]_i$. Let $q(\alpha_l)$ be the density of α_l and impose a prior $p(x_i) = \mathrm{CN}(x_i; 0, \sigma^2_{x_i})$ on x_i. The LMMSE estimator of x_i is now

$$\hat{x}_i^{\mathrm{LMMSE}} = \frac{y_i \langle h_i \rangle^*_{\boldsymbol{\alpha},\boldsymbol{\tau}}}{\left\langle |h_i|^2 \right\rangle_{\boldsymbol{\alpha},\boldsymbol{\tau}} + \hat{\beta}\sigma^{-2}_{x_i}}.$$

By letting $\sigma^2_{x_i} \to \infty$ to express that we have no prior information on x_i, we recover the mean in (A.32). Note that a similar analogy does not exists for the variance in (A.32).

All remaining messages passed in the BP subgraph, are functions of discrete variables (i.e., coded or information bits). These messages are calculated with the sum-product algorithm, see e.g. [49, 50]. Due to space constraints, we do not give the details here.

When BP messages have been passed in the BP subgraph, the beliefs of the data symbols x_i, $i \in \mathcal{D}$, are calculated from

$$q(x_i) \propto m^{\mathrm{MF}}_{f_{D_i} \to x_i}(x_i) m^{\mathrm{BP}}_{f_{M_i} \to x_i}(x_i). \tag{A.33}$$

Since $q(x_i)$ is a probability mass function, we can use straightforward evaluation of finite sums to obtain $\langle \mathbf{X} \rangle_{\boldsymbol{x}_{\mathcal{D}}}$ and $\langle \mathbf{X}^{\mathrm{H}} \mathbf{X} \rangle_{\boldsymbol{x}_{\mathcal{D}}}$, which are used in the belief updates in the MF subgraph.

A.4.3 An Incremental Algorithm

Algorithm A.1 combines the derived belief update expressions into a turbo receiver with sparsity-based parametric channel estimation. The algorithm is split into two parts: channel estimation (lines 5 - 30) and decoding (line 32). The outer loop alternates between these two steps until the information bit estimates have not changed in 10 iterations or a maximum of 50 iterations is reached.

The scheduling of the channel estimation is inspired by [31]. The basic idea is to construct a representation of the CFR in (A.7) by sequential refinement of the estimated multipath components. One component is determined by the parameters (z_l, α_l, τ_l) for a particular index l. All multipath components are initialized in the inactivated state, i.e., $\hat{\boldsymbol{z}}$ is the zero vector.

The channel estimation procedure alternates between two stages: In the activation stage (at line 7) one of the inactive components is activated and its multipath delay and coefficient are calculated. The activation criterion (A.25) determines if the component should stay activated. In the second stage (starting at line 20), all active components are sequentially refined. Again, the criterion (A.25) determines if a component should be deactivated. The channel estimation procedure thus iteratively adds, updates and possibly removes components until the stopping criterion is fulfilled. The multipath delays are tracked via the scheme in Sec. A.4.1 in a way that resembles the operation of a rake receiver [17]. The approach presented here differs from that implemented in a rake receiver in that it provides an integral criterion for inclusion or exclusion of components (rake "fingers") via (A.25). The multipath delay of the newly activated component is found via a maximization over the grid $\tilde{\boldsymbol{\tau}}$. The grid should have a sufficiently fine resolution, such that the initial estimate of the delay is close to the maximizer in (A.22). We choose the distance between points in the grid as $(N\Delta_f)^{-1}/8$. As inner stopping criterion we use $|1/\hat{\beta}^{[t]} - 1/\hat{\beta}^{[t-1]}| < 10^{-3}/\hat{\beta}^{[t-1]}$, where t is the inner iteration number. The number of inner iterations is limited to 50.

Algorithm A.1: Parametric BP-MF receiver.

Input: Observations $\boldsymbol{y}$, pilot indices $\mathcal{P}$ and pilot symbols $\boldsymbol{x}_{\mathcal{P}}$.
Output: Belief functions of data bits $\{q(u_k)\}_{k\in\mathcal{K}}$.
Notes: Define the set of components as $\mathcal{L} = \{1, \ldots, L_{\max}\}$ and the set of active components as $\hat{\mathcal{A}} \triangleq \{l \in \mathcal{L} : \hat{z}_l = 1\}$.

$\tilde{\boldsymbol{\tau}} \leftarrow$ Vector with values from equispaced grid on $[-\frac{1/2}{N\Delta_f}, T_{\mathrm{CP}}]$.
Initialize channel parameter estimates $(\hat{\rho}, \hat{\eta}, \hat{\beta})$.
$\hat{\boldsymbol{z}}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^2 \leftarrow$ Zero vectors of length N.
while *Outer stopping criterion not met* **do**
    while *Inner stopping criterion not met* **do**
        $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}, \hat{\boldsymbol{\sigma}}^2_{\hat{\mathcal{A}}} \leftarrow$ Updates from (A.31) and (A.13).
        Activate an inactive component:
        if *the inactive set* $\mathcal{L}\backslash\hat{\mathcal{A}}$ *is non-empty* **then**
            $l \leftarrow$ Any index from the inactive set $\mathcal{L}\backslash\hat{\mathcal{A}}$.
            $\hat{z}_l \leftarrow 1$.
            $\hat{\tau}_l \leftarrow$ Value from (A.22) calculated on the grid $\tilde{\boldsymbol{\tau}}$.
            $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}, \hat{\boldsymbol{\sigma}}^2_{\hat{\mathcal{A}}} \leftarrow$ Updates from (A.31) and (A.13).
            $\hat{\tau}_l \leftarrow$ Update via the scheme in Sec. A.4.1.
            $\hat{\mu}_l, \hat{\sigma}^2_l \leftarrow$ Updates from (A.12) and (A.13).
            if *activation criterion* (A.25) *is false* **then**
                $\hat{z}_l \leftarrow 0$.
                Reset $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}$ to the value calculated in line 6.
            end
        end
        Update all components currently included in model:
        for $l \in \hat{\mathcal{A}}$ **do**
            $\hat{\tau}_l \leftarrow$ Update via the scheme in Sec. A.4.1.
            $\hat{\mu}_l, \hat{\sigma}^2_l \leftarrow$ Updates from (A.12) and (A.13).
            if *activation criterion* (A.25) *is false* **then**
                $\hat{z}_l \leftarrow 0$.
            end
        end
        $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}, \hat{\boldsymbol{\sigma}}^2_{\hat{\mathcal{A}}} \leftarrow$ Updates from (A.31) and (A.13).
        $\hat{\rho}, \hat{\eta}, \hat{\beta} \leftarrow$ Updates from (A.28), (A.29) and (A.30).
    end
    Update the messages $m^{\mathrm{MF}}_{f_{\mathcal{D}_i}\to x_i}(x_i)$ from (A.32).
    Iterate message-passing in the BP subgraph.
    Update the beliefs $q(x_i)$ from (A.33).
end

During the first outer iteration the decoder has not been used yet and symbol beliefs $q(x_i)$ of the data subcarriers (indices $i \in \mathcal{D}$) are not available. During the first iteration the channel estimator therefore only uses the pilot subcarriers (indices $j \in \mathcal{P}$). To avoid any identifiability issue regarding the multipath delays (see Sec. A.5.3) during the pilot-only iteration, the multipath delays estimated in this iteration are restricted to the interval $[0, 1/(\Delta_f \Delta_P))$, where Δ_P is the pilot spacing.[8]

The active component prior variance is initialized to $\hat{\eta} = 1$ and the activation probability is initialized to $\hat{\rho} = 0.5$. We initialize the noise variance to $\hat{\beta} = ||\boldsymbol{y}||^2/N \cdot 10^{-15/10}$ (i.e., assuming approximately 15 dB SNR). The activation probability and noise variance is kept fixed during the first 3 outer iterations, because these can only be accurately estimated when a reliable estimate of the channel is available.

A.4.4 Convergence Analysis and Computational Complexity

We now wish to analyze the convergence properties of Algorithm A.1. First recognize that the algorithm alternates between updates in the MF and BP subgraphs of Fig. A.2. To analyze convergence, we discuss under which conditions each of these sets of updates are guaranteed not to increase the RBFE. If all updates give a non-increasing RBFE it can be concluded that the algorithm converges, since the RBFE is bounded below.

We first discuss the updates in the MF subgraph, i.e., of belief functions $q(\alpha_l)$ ($l \in \mathcal{L}$) and point estimates $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\tau}}, \hat{\rho}, \hat{\eta}, \hat{\beta})$. During these updates the messages $m^{\text{BP}}_{f_{M_i} \to x_i}(x_i)$ are kept fixed. The joint update of $\hat{\tau}_l$ and $q(\alpha_l)$ gives a non-increasing RBFE as per Lemma A.1. A similar conclusion can be drawn regarding the joint update of $\hat{z}_l$ and $q(\alpha_l)$. The individual update of $q(\alpha_l)$ is found via the method of Lagrange multipliers applied to the RBFE with normalization constraint $\int q(\alpha_l)\,\mathrm{d}\alpha_l = 1$. The second-order functional derivative of the RBFE $\frac{\delta^2 F_{\text{BP-MF}}}{\delta q^2(\alpha_l)} = \frac{1}{q(\alpha_l)}$ is a positive semi-definite function; it follows that the RBFE is convex in this argument. It can be concluded that the update of $q(\alpha_l)$ is the global minimizer of the RBFE and the objective is thus non-increasing. A similar conclusion can be drawn regarding the update of the channel parameters, cf. Eq. (A.26). All updates in the MF subgraph thus give non-increasing RBFE.

We now analyze the convergence in the BP subgraph, i.e., the updates of belief functions $q(x_i), q([\boldsymbol{c}^{(i)}]_q)$ and $q(u_k)$. Considering the belief functions of variables in the MF subgraph as fixed and ignoring scaling and constant terms, the RBFE is equal to the Bethe free energy corresponding to the fac-

[8] We define the pilot spacing as $\Delta_P = D + 1$, where D is the number of data subcarriers between any two neighboring pilot subcarriers.

torization (see [8, Appendix E])

$$\prod_{i\in\mathcal{D}} m^{\mathrm{MF}}_{f_{D_i}\to x_i}(x_i)p(x_i|\boldsymbol{c}^{(i)})p(\boldsymbol{c}|\boldsymbol{u})\prod_{k\in\mathcal{K}} p(u_k).$$

Further, all messages in the BP subgraph are equal to the messages obtained from BP applied to the above factorization. This means that we can analyze the behaviour of message-passing in the BP subgraph, by analyzing BP applied to the above factorization. If the factor graph does not contain any cycles it can be shown that BP globally minimizes the Bethe free energy [8, 45] (which in this case is equal to the Gibss free energy) and convergence of the complete BP-MF receiver algorithm is guaranteed. Recall that the factor $f_C(\boldsymbol{c},\boldsymbol{u}) = p(\boldsymbol{c}|\boldsymbol{u})$ describes the channel code and may be replaced by a number of auxiliary variables and factors. The specific structure of the BP factor graph is thus determined by the channel code. In the special case of convolutional coding with binary or quadrature phase-shift keying (BPSK or QPSK) modulation, the BP graph does indeed become a tree-graph and convergence of Alg. A.1 is guaranteed. If the modulation order is higher than QPSK, loops occur between f_{M_i} and f_C and convergence can thus not be guaranteed.

For other common channel codes, such as Turbo and LDPC codes, the subgraph represented by f_C contains loops. However, BP has empirically been shown to converge for decoding of many channel codes and it is a well known practice to use BP even though convergence cannot be guaranteed theoretically, see e.g. [49–52]. When BP does converge it has been shown to be to a (local) minimum of the Bethe free energy [54], which further explains why we do indeed obtain convergence of Alg. A.1 in our numerical investigations. Conditions exist under which BP is guaranteed to converge in loopy graphs, e.g. [55, 56]. These are, however, not applicable to our situation.

We now turn our attention to the computational complexity of the channel estimator, i.e., the loop starting at line 5. The most demanding part of the channel estimation in terms of computational complexity is the calculation of $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}$ via (A.31). We show in Appendix B that (under a conjecture) this update can be calculated in time $\mathcal{O}\Big(\min(\hat{L}^2N, \hat{L}N\sqrt{N})\Big)$, where $\hat{L}$ is the number of components currently included in the model.

The grid search in line 11 is recognized as the maximization of the periodogram, which can be calculated via a fast Fourier transform in time $\mathcal{O}(N\log N)$ when the grid is assumed to be of size $\mathcal{O}(N)$.

The loop starting at line 21 necessitates the calculation of $\boldsymbol{r}$ in (A.16). Direct computation has complexity $\mathcal{O}(\hat{L}N)$ for each of the $\hat{L}$ iterations in the loop. By updating $\boldsymbol{r}$ with each change to $\hat{\boldsymbol{\mu}}$, the direct evaluation can be avoided and the complexity of each iteration in the loop becomes $\mathcal{O}(N)$, which is the same as that of all other operations inside the loop. The overall complexity of the loop is thus $\mathcal{O}(\hat{L}N)$.

With these remarks, we see that the overall complexity per iteration of the channel estimator is $\mathcal{O}\Big(\min(\hat{L}^2 N, \hat{L} N \sqrt{N})\Big)$.

A.5 Numerical Evaluation

In our numerical evaluation we consider an OFDM system as described in Sec. A.2. We use a random interleaver and a rate–1/2 non-systematic convolutional channel code, decoded by the loopy BP implementation from the Coded Modulation Library.[9] The pilot signals are chosen at random from a QPSK alphabet. The first and last subcarriers are designated as pilots. The other pilot subcarriers are located equispaced with spacing Δ_P, i.e., the number of data subcarriers between two such neighbour pilot subcarriers is $\Delta_P - 1$.[10] The SNR is defined based on the realization of the channel frequency response as

$$\mathrm{SNR} \triangleq \frac{\mathbb{E}\left[|x_i|^2\right] ||\boldsymbol{h}||^2}{N\beta}, \tag{A.34}$$

where $\mathbb{E}\left[|x_i|^2\right]$ is calculated under the assumption that the symbols in the respective alphabets $\mathbb{A}_\mathrm{D}$ and $\mathbb{A}_\mathrm{P}$ are equiprobable.

We asses how the receivers behave in two different scenarios. The parameters considered in each scenario are listed in Table A.1. Scenario A uses the channel model put forward by ITU for the evaluation of IMT-Advanced radio interface technologies [12]. Specifically we use the model with the parameter setting for urban macro (UMa) environment with non line-of-sight (NLOS) conditions. The model generates CIRs typical of macro-cellular communication in an urban environment targeting continuous coverage for pedestrian up to fast vehicular users [12]. The channel model [12] is specified for use with up to 100 MHz bandwidth, while the system we are simulating uses 25.6 Mhz bandwidth. We are thus well within the specified bandwidth range.

Scenario B uses the standardized model proposed for the evaluation of IEEE 802.15.4a UWB technologies [13]. Specifically we use the model with the setting proposed for outdoor environments with NLOS conditions. The model generates CIRs typical of micro-cellular communication in a suburban-like environment with a rather small range [13]. Note that this model is also used in [11].

Since our signal model (A.5) is not valid for CIRs longer than the cyclic prefix duration T_CP, we drop realizations with component delays larger than T_CP. Fig. A.3 shows 3 CIR realizations generated for each of scenarios A and

[9] Available from http://iterativesolutions.com/Matlab.htm

[10] We have also conducted experiments with random pilot patterns (not shown), but have seen no significant benefit in doing so for the setup considered here.

Parameter	Scenario A	Scenario B
Channel model	ITU-R M.2135 UMa NLOS [12]	IEEE 802.15.a Outdoor NLOS [13]
Number of subcarriers (N)	1024	1024
Modulation format of data subcarriers	256-QAM	256-QAM
Convolutional code polynomial	$(561, 753)_8$	$(561, 753)_8$
Subcarrier spacing (Δ_f)	25 kHz	250 kHz
Cyclic prefix duration (T_{CP})	5200 ns	800 ns
Number of equispaced pilots	172	256
Pilot spacing (Δ_P) (implied by the above)	6	4

Table A.1: Simulation parameters.

B, along with an estimate of the power-delay profile (PDP). We observe that in Scenario A the CIRs show a specular channel behaviour ($L \ll \lceil T_{\mathrm{CP}} N \Delta_f \rceil$), while they show a diffuse behaviour ($L \gg \lceil T_{\mathrm{CP}} N \Delta_f \rceil$) in Scenario B.

We asses the performance of the considered receivers in terms of average coded bit error rate (BER) and normalized mean squared error (MSE) of the CFR, calculated as $||\hat{\boldsymbol{h}} - \boldsymbol{h}||^2 / ||\boldsymbol{h}||^2$. These averages are obtained from 500 Monte Carlo trials ($\approx 1.5 \cdot 10^6$ information bits) for SNR $< 20\,\mathrm{dB}$, with one OFDM symbol transmitted in each trial. To get reliable BER estimates we use $3,000$ trials ($\approx 10^7$ information bits) for SNR $= 20\,\mathrm{dB}$ and $15,000$ trials ($\approx 4.5 \cdot 10^7$ information bits) for SNR $> 20\,\mathrm{dB}$. The OFDM symbols and channel realizations are generated i.i.d. according to the above.

A.5.1 Evaluation and Comparison with Other Algorithms

We evaluate our algorithm (Parametric BP-MF) and compare with the following reference algorithms:

Turbo-GAMP [11]: The algorithm employs a baud-spaced grid in the delay domain, i.e., the resolution of the grid is $T_s = (N\Delta_f)^{-1} \approx 39\,\mathrm{ns}$ for Scenario A and $T_s \approx 3.9\,\mathrm{ns}$ for Scenario B. For each channel tap a large-tap and small-tap variance is provided along with tap-state transition probabilities (see [11] for more details). These are estimated via the EM algorithm provided in [11] from $50,000$ channel realizations. Turbo-GAMP is provided with significant prior information on the CIR via these statistical values. We also provide Turbo-GAMP with the true noise variance, as [11] does not give a way to estimate this value.

LMMSE BP-MF [9]: The algorithm directly estimates the CFR $\boldsymbol{h}$ via the

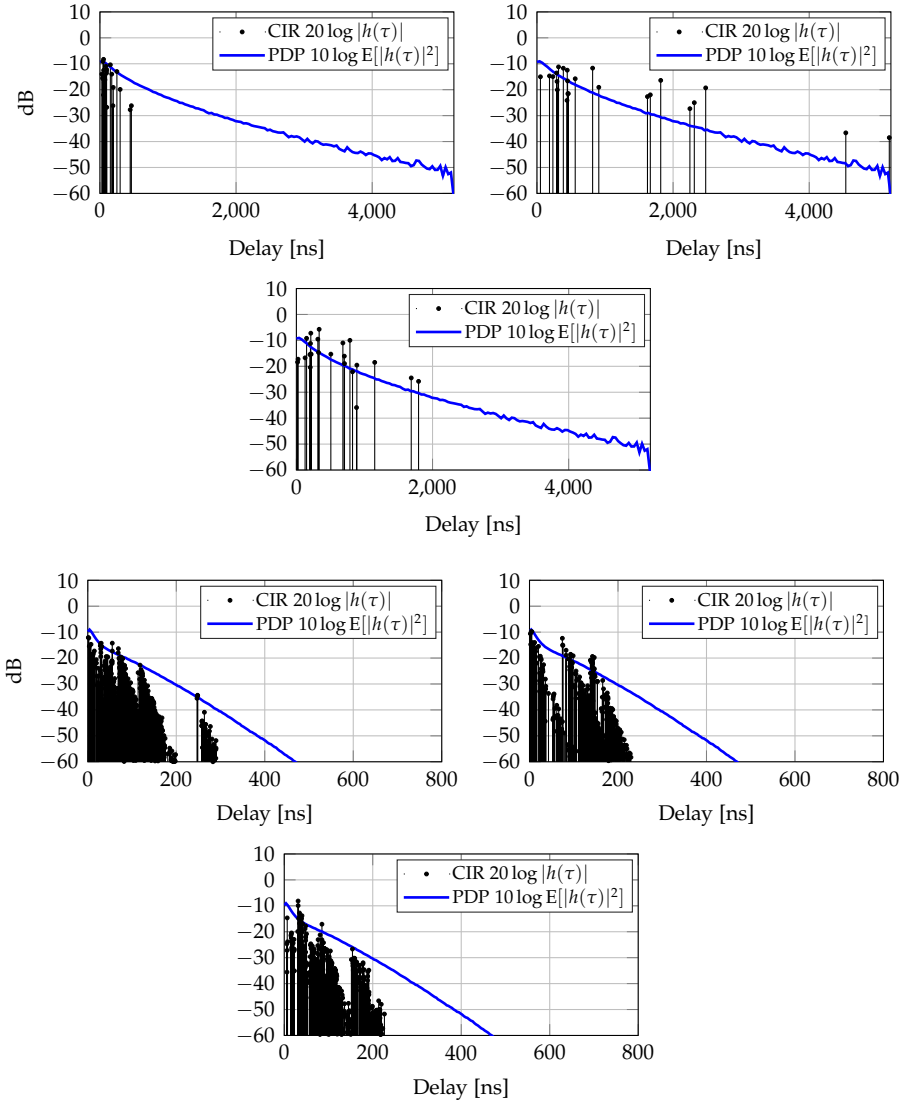

Fig. A.3: Three sample realizations of the CIR for Scenario A (top) and Scenario B (bottom). An estimate of the power-delay profile is also shown, which is obtained by averaging the magnitude-squared impulse responses of 50,000 channel realizations.

BP-MF framework. It is a turbo receiver with LMMSE channel estimation which requires prior knowledge of the noise variance and the covariance matrix $\mathbb{E}\left[\boldsymbol{h}\boldsymbol{h}^{\mathrm{H}}\right]$ (or, equivalently, the channel PDP). We provide the true noise variance to the receiver and show results using covariance matrices calculated based on *a)* the robust PDP assumption described in [36], which assumes constant PDP within the interval $[0, T_{\mathrm{CP}})$, and *b)* with knowledge of the true covariance matrix. Due to the complex structure of the channel models, the true covariance matrix is not easy to obtain analytically. We therefore estimate it as the sample covariance matrix obtained from $50,000$ channel realizations. Similarly we compute an estimate of the PDP which is shown in Fig. A.3. We identify these estimates with their true counterpart.

Perfect CSI: This oracle receiver has perfect channel state information (CSI), i.e., it knows the true CFR $\boldsymbol{h}$, and thus provides a lower bound on the achievable BER. The Perfect CSI trace is only shown in the BER plots. To be specific, it is implemented by computing the messages $n_{x_i \to f_{M_i}}(x_i)$ for all $i \in \mathcal{D}$ (see (A.32)), followed by 5 iterations in the BP subgraph of Fig. A.2.

Oracle LMMSE: This trace is only shown for channel MSE. It is obtained as the LMMSE estimate of $\boldsymbol{h}$ with the knowledge of the noise variance β, transmitted symbol vector $\boldsymbol{x}$ (i.e., both pilots and data are known), the vector of delays $\boldsymbol{\tau}$ and the probability density function of the channel multipath coefficients in $\boldsymbol{\alpha}$, i.e., the probability density function of $\boldsymbol{h}$ is known exactly. In the channel model of Scenario A, the entries in $\boldsymbol{\alpha}$ are complex normal and the LMMSE estimator achieves the MMSE. In Scenario B the multipath coefficients in $\boldsymbol{\alpha}$ are not normal. However, by a central limit theorem argument, the vector $\boldsymbol{h}$ is to a good approximation also normal in Scenario B and again the LMMSE estimator achieves the MMSE.

A.5.2 Varying the Signal-to-Noise Ratio

Fig. A.4 shows performance results for varying SNR in Scenario A. We first note that Parametric BP-MF performs very well in both BER and MSE. Its BER is remarkably close to that of the Perfect CSI trace, indicating that there is very little margin for improvement of the algorithm in this scenario. Both versions of LMMSE BP-MF (known and robust PDP) show higher BER than Parametric BP-MF, corresponding to a decrease in SNR of about 1 dB. They show almost the same BER performance because the delay spread in Scenario A is relatively large (cf. Fig. A.3) and the robust PDP assumption is therefore realistic. Turbo-GAMP does not perform well and shows a BER floor at high SNR. The reason is discussed below.

Fig. A.5 shows the corresponding results for Scenario B. We here observe that Parametric BP-MF has a BER loss compared to the Perfect CSI trace corresponding to about 0.5 dB SNR difference. Parametric BP-MF is among the best performing algorithms, even though the CIRs generated in Scenario B

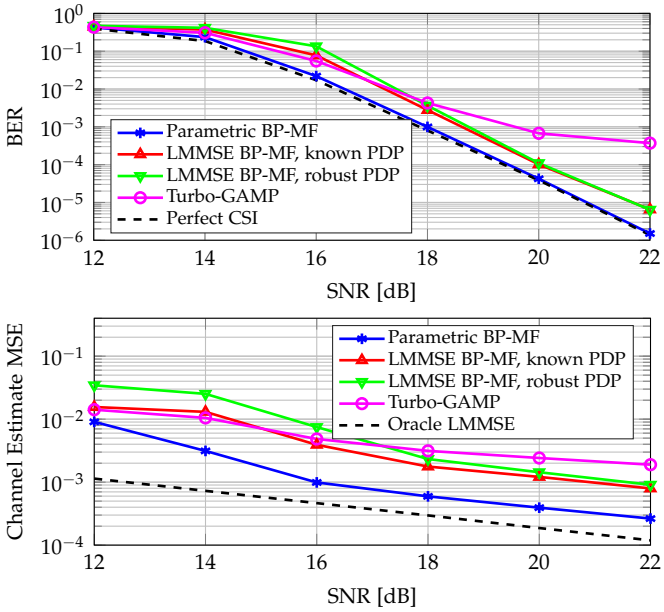

Fig. A.4: BER (top) and MSE of CFR estimate (bottom) vs. SNR in Scenario A.

are composed of a very large number of multipath components that the algorithm cannot resolve individually (cf. Fig. A.3). Instead, the algorithm estimates a CIR with significantly fewer components of which the corresponding CFR approximates the true channel response within the system bandwidth. We have observed that the estimated CIR "recovers the support" of the true CIR, in the sense that an estimated multipath component is located wherever the CIR contains significant power. Parametric BP-MF has a BER and MSE performance equivalent to that of LMMSE BP-MF with known PDP. We stress that Parametric BP-MF achieves this performance without using prior knowledge of the channel. Since the true PDP is not known in a real propagation environment, the LMMSE BP-MF receiver with known PDP cannot readily be realized in practice.

In Scenario B we observe a significant difference between the LMMSE BP-MF algorithms with known and robust PDP. To explain this difference observe in Fig. A.3 that most of the mass of the PDP is located at small delays. This significantly deviates from the evenly distributed mass on $[0, T_{\mathrm{CP}})$ that underlies the robust PDP assumption.

In both Fig. A.4 and Fig. A.5 an error floor is observed for Turbo-GAMP at high SNR.[11] We conjecture that this error floor is caused by the restriction of

[11] In [11], where Turbo-GAMP is introduced, such an error floor is not observed even though the setup in the numerical investigation is almost identical to that in Scenario B. The reason is

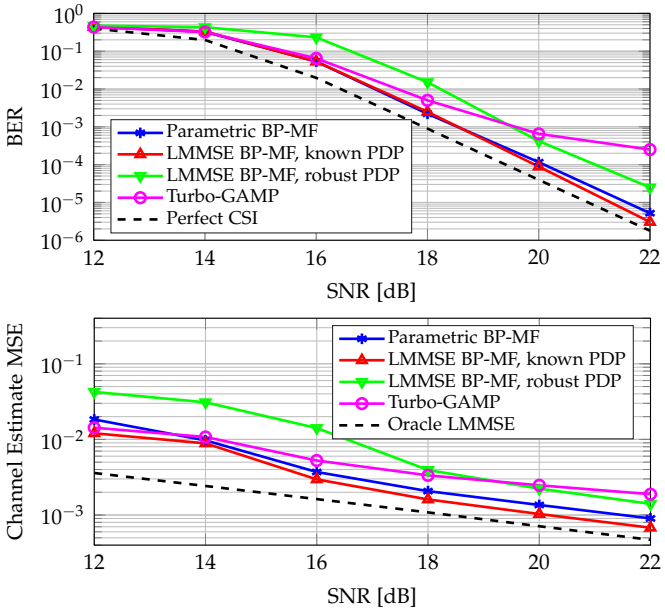

Fig. A.5: BER (top) and MSE of CFR estimate (bottom) vs. SNR in Scenario B.

the delays to the baud-spaced grid. If the delays are generated to be located on that grid, the performance of Turbo-GAMP is very close to that of the Perfect CSI trace (not shown here). We have also conducted experiments with random pilot patterns (not shown) as used in [11] (where Turbo-GAMP is introduced) but did not see an improvement of Turbo-GAMP in that case. We note that such error floors in BER and MSE have previously been observed for other grid-based sparse channel estimation algorithms, see for example [21, 40]. In conclusion, the grid-based approximation is of insufficient accuracy for communication with large modulation order in the high-SNR regime.

A.5.3 Varying the Number of Pilots

We now investigate if our receiver design improves the trade-off between the number of pilots and estimator performance. To do so, we refer to Fig. A.6 and A.7, which show the BER performance for varying number of pilot subcarriers.

The first observation is that LMMSE BP-MF with robust PDP shows a

an error in the signal model in [11], which invalidates the numerical results obtained in that paper. Specifically the error occurs when the "uniformly sampled channel taps" are defined as rate $1/T$ samples of the compound CIR $x(\tau) \triangleq (g_r * h * g_t)(\tau)$. However, since $(g_r * g_t)(\tau)$ is a raised-cosine filter with design parameter 0.5, $x(\tau)$ has bandwidth $1.5/T$, leading to aliasing in the sampling operation.

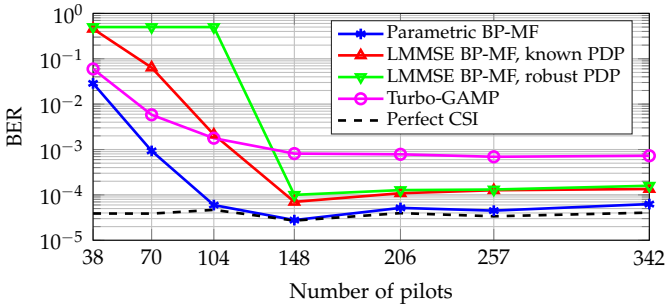

Fig. A.6: BER vs. number of pilot subcarriers in Scenario A at 20 dB SNR.

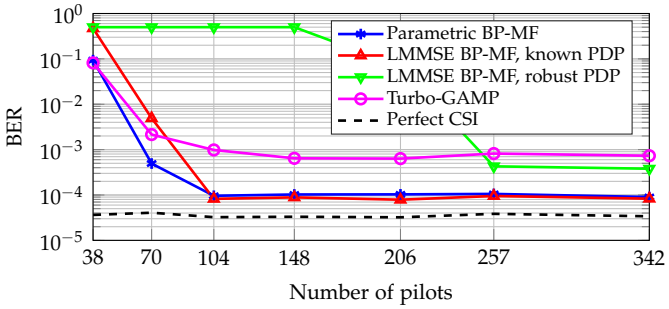

Fig. A.7: BER vs. number of pilot subcarriers in Scenario B at 20 dB SNR.

point at which the BER performance quickly transitions between high (50 %) BER and low ($< 10^{-3}$) BER. Under the robust PDP assumption, the channel coherence bandwidth is approximately $1/T_{\text{CP}}$. As a rule of thumb there should be at least one pilot subcarrier per coherence interval, which gives the criterion $P > N\Delta_f T_{\text{CP}}$, where P is the needed number of pilot subcarriers. For Scenario A we have $N\Delta_f T_{\text{CP}} \approx 133$ and for Scenario B we have $N\Delta_f T_{\text{CP}} \approx 205$, which exactly are the respective numbers of pilots at which LMMSE BP-MF with robust PDP transitions between low and high BER.

All algorithms except LMMSE BP-MF with robust PDP can operate significantly below the above-mentioned limit. Due to the turbo structure, the number of pilots can be decreased significantly without incurring an increase in BER. Parametric BP-MF is the only algorithm which performs well with 104 pilot subcarriers in both scenarios. LMMSE BP-MF with known PDP is close to achieving the same performance as Parametric BP-MF, but we reiterate that this receiver requires prior knowledge of the PDP.

A.6 Conclusions

In this paper we proposed a turbo OFDM receiver which employs sparsity-based parametric channel estimation. The iterative receiver is derived using the BP-MF framework for approximate Bayesian inference. Unlike state-of-the-art sparse channel estimators, our scheme does not restrict multipath delays of the estimated channel impulse response to a grid. As a result it can truly exploit parsimony of the channel impulse response, without resorting to approximate sparsity (as in [11, 23, 24]).

We have presented a numerical evaluation that compares our algorithm with state-of-the-art methods, i.e., Turbo-GAMP [11] and LMMSE BP-MF [9]. This study demonstrated that restricting the multipath delays to a baud-spaced grid (e.g., as in Turbo-GAMP) is not a viable approach because the resulting equivalent vector of channel taps is only approximately sparse.

The numerical evaluation also shows that our proposed scheme can effectively exploit the structure of wireless channels. Even though our receiver is derived assuming the channel to be specular (as in Scenario A), it also performs very well in diffuse channels (Scenario B). Thus, we expect that it will also perform well in any channel which shows a mixture of both characteristics.

A The Region-Based Free Energy Approximation

At the heart of the derivation of our algorithm lies the RBFE as defined by [8, Eq. (17)], [45]. In this paper we use the RBFE of the probability distribution corresponding to the factor graph depicted in Fig. A.2. For convenience, we give here the complete expression of the RBFE:

$$F_{\text{BP-MF}} = F_{\text{BP}} + F_{\text{MF}} \tag{A.35}$$

with

$$\begin{aligned} F_{\text{BP}} = &\sum_{k\in\mathcal{K}} \sum_{u_k\in\{0,1\}} b_{u_k}(u_k) \ln \frac{b_{u_k}(u_k)}{p(u_k)} + \sum_{i\in\mathcal{D}} \sum_{\substack{x_i\in\mathbb{A}_{\text{D}} \\ \boldsymbol{c}^{(i)}\in\{0,1\}^Q}} b_{M_i}(x_i,\boldsymbol{c}^{(i)}) \ln \frac{b_{M_i}(x_i,\boldsymbol{c}^{(i)})}{p(x_i|\boldsymbol{c}^{(i)})} \\ &+ \sum_{\substack{\boldsymbol{c}\in\{0,1\}^{K/R} \\ \boldsymbol{u}\in\{0,1\}^K}} b_C(\boldsymbol{c},\boldsymbol{u}) \ln \frac{b_C(\boldsymbol{c},\boldsymbol{u})}{p(\boldsymbol{c}|\boldsymbol{u})} - \sum_{k\in\mathcal{K}} \sum_{u_k\in\{0,1\}} q(u_k) \ln q(u_k) \\ &- \sum_{i\in\mathcal{D}} \sum_{m\in\{1,\ldots,Q\}} \sum_{[\boldsymbol{c}^{(i)}]_m\in\{0,1\}} q([\boldsymbol{c}^{(i)}]_m) \ln q([\boldsymbol{c}^{(i)}]_m), \end{aligned}$$

$$F_{\text{MF}} = \sum_{l\in\mathcal{L}} \int q(\alpha_l)\ln q(\alpha_l)\,\mathrm{d}\alpha_l - \left\langle \ln p(\boldsymbol{y}|\boldsymbol{x}_{\mathcal{D}},\boldsymbol{\alpha},\hat{\boldsymbol{\tau}};\hat{\beta})\right\rangle_{\boldsymbol{x}_{\mathcal{D}},\boldsymbol{\alpha}} - \sum_{l\in\mathcal{L}} \left\langle \ln p(\alpha_l|\hat{z}_l;\hat{\eta})p(\hat{z}_l;\hat{\rho})p(\hat{\tau}_l)\right\rangle_{\alpha_l},$$

where $b_C(\boldsymbol{c},\boldsymbol{u})$, $b_{M_i}(x_i,\boldsymbol{c}^{(i)})$ for $i\in\mathcal{D}$ and $b_{u_k}(u_k)$ for $k\in\mathcal{K}$ are factor beliefs. With abuse of notation we let $q(\cdot)$ denote variable beliefs and $\langle\cdot\rangle_a$ denote expectation with respect to the belief density $q(a)$.

B Efficient Calculation of $\hat{\mu}_{\mathcal{A}}$ When $\hat{L}$ is Large

In this appendix we present a computationally efficient method for evaluating $\hat{\boldsymbol{\mu}}_{\hat{\mathcal{A}}}$ as defined by (A.31). We first note that direct evaluation and inversion of $\mathbf{Q}$ has time complexity $\mathcal{O}(\hat{L}^2N)$, where $\hat{L}\triangleq|\hat{\mathcal{A}}|$. The method is iterative and has complexity $\mathcal{O}(\hat{L}N\sqrt{N})$ provided Conjecture A.1 (below) holds. It is thus beneficial to use it when $\hat{L}$ grows faster than $\sqrt{N}$.

We first use the Woodbury matrix identity to write $\hat{\boldsymbol{\mu}}$ as

$$\hat{\boldsymbol{\mu}} = \hat{\beta}^{-1}\hat{\eta}\left(\mathbf{I}-\hat{\beta}^{-1}\hat{\eta}\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\mathbf{C}^{-1}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\right)\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\left\langle\mathbf{X}\right\rangle_{\boldsymbol{x}_{\mathcal{D}}}^{\mathrm{H}}\boldsymbol{y},$$

where

$$\mathbf{C} = \left\langle\mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\boldsymbol{x}_{\mathcal{D}}}^{-1} + \hat{\beta}^{-1}\hat{\eta}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}}).$$

We immediately recognize that the computationally dominating part is to solve a system of N linear equations of the form $\mathbf{C}\boldsymbol{z}=\boldsymbol{a}$. Since $\mathbf{C}$ is Hermitian and positive-definite, we can solve this system via the conjugate-gradient (CG) method (Alg. 2.1 in [57]), which is an iterative method for solving systems of linear equations. In the following we show that the number of iterations of the CG method is $\mathcal{O}(\sqrt{N})$.

We first need a conjecture on the eigenvalues of the (Hermitian-Toeplitz) matrix $\mathbf{T}=\hat{\beta}^{-1}\hat{\eta}\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\tau}}_{\hat{\mathcal{A}}})$.

Conjecture A.1. *There exists an upper bound on the largest eigenvalue of* $\mathbf{T}$ *which grows linearly with* N*, i.e.,*

$$\lambda_{\max}(\mathbf{T}) = \mathcal{O}(N).$$

To justify this conjecture we refer to Fig. A.8, where the largest eigenvalue is shown for varying N.

We also need a number of lemmas.

Lemma A.2. *There exists constants* $c_1>0$ *and* $c_2<\infty$*, such that* $c_1\le\left\langle|x_i|^2\right\rangle_{\boldsymbol{x}_{\mathcal{D}}}\le c_2$ *for all* $i\in\mathcal{D}\cup\mathcal{P}$*.*

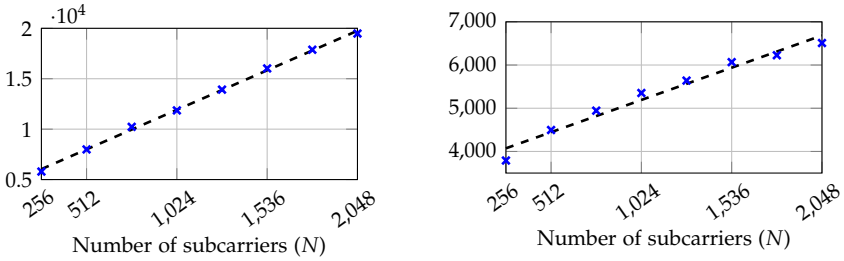

Fig. A.8: Average of the largest eigenvalue of the matrix $\mathbf{T}$ encountered during one execution of Parametric BP-MF for Scenario A (left) and B (right). Average obtained from 1000 Monte-Carlo trials. Both plots were generated at 20 dB SNR. A dashed line depicts the least-squares linear fit.

Proof: Observe that the data and pilot modulation symbol alphabets $\mathbb{A}_\mathrm{D}$ and $\mathbb{A}_\mathrm{P}$ only contain finite, non-zero values. We can thus take $c_1 = \min_{x\in\mathbb{A}_\mathrm{P}\cup\mathbb{A}_\mathrm{D}} |x|^2$ and $c_2 = \max_{x\in\mathbb{A}_\mathrm{P}\cup\mathbb{A}_\mathrm{D}} |x|^2$ to complete the proof. ■

Lemma A.3. *Assume that Conjecture A.1 holds. The largest and smallest eigenvalues of* $\mathbf{C}$ *obey*

$$\lambda_{\max}(\mathbf{C}) = \mathcal{O}(N), \qquad \lambda_{\min}(\mathbf{C}) \geq c_1^{-1}.$$

Proof: By the Weyl inequality for Hermitian matrices $\mathbf{C}$, $\mathbf{T}$ and $\left\langle \mathbf{X}^\mathrm{H}\mathbf{X} \right\rangle_{\boldsymbol{x}_\mathcal{D}}^{-1}$ we have

$$\lambda_{\max}(\mathbf{C}) \leq \lambda_{\max}\left(\left\langle \mathbf{X}^\mathrm{H}\mathbf{X} \right\rangle_{\boldsymbol{x}_\mathcal{D}}^{-1} \right) + \lambda_{\max}(\mathbf{T}).$$

The first inequality follows directly from Conjecture A.1 and Lemma A.2.

Similarly by the dual Weyl inequality

$$\lambda_{\min}(\mathbf{C}) \geq \lambda_{\min}\left(\left\langle \mathbf{X}^\mathrm{H}\mathbf{X} \right\rangle_{\boldsymbol{x}_\mathcal{D}}^{-1} \right) + \lambda_{\min}(\mathbf{T}).$$

Since $\hat{L} < N$, the matrix $\mathbf{T}$ is singular and $\lambda_{\min}(\mathbf{T}) = 0$. The second inequality now follows from Lemma A.2. ■

By Theorem 2.2 in [57] the number of iterations required by the CG method to achieve a desired accuracy in the solution of $\boldsymbol{a} = \mathbf{C}\boldsymbol{z}$ is $\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{C})}{\lambda_{\min}(\mathbf{C})}} \right)$. By Lemma A.3 the number of iterations is thus $\mathcal{O}(\sqrt{N})$. Each iteration has time complexity $\mathcal{O}(\hat{L}N)$ and the overall complexity of solving (A.31) via this method is therefore $\mathcal{O}(\hat{L}N\sqrt{N})$.

References

[1] *Evolved Universal Terrestrial Radio Access (E-UTRA), Physical channels and modulation*, 3GPP Std. 36.211, Rev. 12.8.0, Dec. 2015.

[2] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, 1995.

[3] M. Tüchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 920–952, Feb. 2011.

[4] S. Park, Y. G. Kim, and C. G. Kang, "Iterative receiver for joint detection and channel estimation in OFDM systems under mobile radio channels," *IEEE Trans. Veh. Technol.*, vol. 53, no. 2, pp. 450–460, Mar. 2004.

[5] A. P. Worthen and W. E. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

[6] S. Wu, L. Kuang, Z. Ni, J. Lu, D. David Huang, and Q. Guo, "Expectation propagation approach to joint channel estimation and decoding for OFDM systems," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 2014, pp. 1941–1945.

[7] Y. Zhu, D. Guo, and M. L. Honig, "A message-passing approach for joint channel estimation, interference mitigation, and decoding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 6008–6018, Dec. 2009.

[8] E. Riegler, G. E. Kirkelund, C. N. Manchón, M. Badiu, and B. H. Fleury, "Merging belief propagation and the mean field approximation: A free energy approach," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013.

[9] M.-A. Badiu, G. E. Kirkelund, C. N. Manchón, E. Riegler, and B. H. Fleury, "Message-passing algorithms for channel estimation and decoding using approximate inference," in *Proc. IEEE Int. Symp. Inform. Theory*, Jul. 2012, pp. 2376–2380.

[10] P. Schniter, "Belief-propagation-based joint channel estimation and decoding for spectrally efficient communication over unknown sparse channels," *Physical Communication*, vol. 5, no. 2, pp. 91–101, Jun. 2012.

[11] ——, "A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels," *IEEE J. Select. Topics in Signal Process.*, vol. 5, no. 8, pp. 1462–1474, Dec. 2011.

[12] *Guidelines for evaluation of radio interface technologies for IMT-Advanced*, ITU-R Std. M.2135-1, Dec. 2008.

[13] A. F. Molisch, D. Cassioli, C.-C. Chong, S. Emami, A. Fort, B. Kannan, J. Karedal, J. Kunisch, H. G. Schantz, K. Siwiak *et al.*, "A comprehensive standardized model for ultrawideband propagation channels," *IEEE Trans. Antennas Propag.*, vol. 54, no. 11, pp. 3151–3166, Nov. 2006.

[14] M. Feder and J. A. Catipovic, "Algorithms for joint channel estimation and data recovery-application to equalization in underwater communications," *IEEE J. Ocean. Eng.*, vol. 16, pp. 42–55, Jan. 1991.

[15] V. Lottici, A. D'Andrea, and U. Mengali, "Channel estimation for ultrawideband communications," *IEEE J. Sel. Areas Commun.*, vol. 20, pp. 1638–1645, Dec. 2002.

[16] I. Maravic, J. Kusuma, and M. Vetterli, "Low-sampling rate UWB channel characterization and synchronization," *J. of Commun. and Networks*, vol. 5, pp. 319–327, Dec. 2003.

[17] A. F. Molisch, A. Mammela, and D. P. Taylor, Eds., *Wideband Wireless Digital Communication*. Prentice Hall, 2000.

[18] W. Bajwa, A. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, Jun. 2010.

[19] J. Paredes, G. Arce, and Z. Wang, "Ultra-wideband compressed sensing: Channel estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 3, pp. 383–395, Oct. 2007.

[20] C. Berger, S. Zhou, J. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708 – 1721, Mar. 2010.

[21] G. Taubock, F. Hlawatsch, D. Eiwen, and H. Rauhut, "Compressive estimation of doubly selective channels in multicarrier syst.: Leakage effects and sparsity-enhancing process." *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 255–271, Apr. 2010.

[22] N. L. Pedersen, C. N. Manchón, D. Shutin, and B. H. Fleury, "Application of Bayesian hierarchical prior modeling to sparse channel estimation," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 3487–3492.

[23] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, Jun. 2014.

[24] ——, "Joint channel estimation and data detection in MIMO-OFDM systems: A sparse Bayesian learning approach," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5369–5382, Jun. 2015.

[25] J.-J. Van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Börjesson, "On channel estimation in OFDM systems," in *Proc. IEEE 45th Veh. Technology Conf.*, vol. 2, Jul. 1995, pp. 815–819.

[26] Y. Chi, L. Scharf, A. Pezeshki, and A. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, pp. 2182–2195, May 2011.

[27] E. J. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Commun. Pure and Appl. Math.*, vol. 67, no. 6, pp. 906–956, 2014.

[28] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, Dec. 2013.

[29] S. Pejoski and V. Kafedziski, "Estimation of sparse time dispersive channels in pilot aided OFDM using atomic norm," *IEEE Wireless Commun. Lett.*, vol. 4, pp. 397–400, Apr. 2015.

[30] Y. Barbotin and M. Vetterli, "Fast and robust parametric estimation of jointly sparse channels," *IEEE J. on Emerging and Select. Topics in Circuits and Syst.*, vol. 2, pp. 402–412, Sep. 2012.

[31] T. L. Hansen, M. A. Badiu, B. H. Fleury, and B. D. Rao, "A sparse Bayesian learning algorithm with dictionary parameter estimation," in *Proc. IEEE 8th Sensor Array and Multichannel Signal Process. Workshop*, Jun. 2014, pp. 385–388.

[32] M.-A. Badiu, T. L. Hansen, and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2247–2261, May 2017.

[33] D. Shutin, W. Wang, and T. Jost, "Incremental sparse Bayesian learning for parameter estimation of superimposed signals," in *Proc. 10th Int. Conf. Sampling Theory and Applicat.*, Jul. 2013.

[34] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3809–3822, Jul. 2012.

[35] J. J. Kormylo and J. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 482–488, May 1982.

[36] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Börjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.

[37] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, May 1993, pp. 1064–1070.

[38] R. Gallager, "Low-density parity-check codes," *Inform. Theory, IRE Trans.*, vol. 8, no. 1, pp. 21–28, January 1962.

[39] M. Luise, R. Reggiannini, and G. M. Vitetta, "Blind equalization/detection for OFDM signals over frequency-selective channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1568–1578, Oct. 1998.

[40] O.-E. Barbu, N. L. Pedersen, C. N. Manchón, G. Monghal, C. Rom, and B. H. Fleury, "Sparse channel estimation including the impact of the transceiver filters with application to OFDM," in *Proc. 15th Int. Workshop on Signal Process. Advances in Wireless Commun.*, Jun. 2014, pp. 424–428.

[41] R. J.-M. Cramer, R. A. Scholtz, and M. Z. Win, "Evaluation of an ultra-wide-band propagation channel," *IEEE Trans. Antennas Propag.*, vol. 50, no. 5, pp. 561–570, May 2002.

[42] A. F. Molisch, "Ultrawideband propagation channels – theory, measurement, and modelling," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1528–1545, Oct. 2004.

[43] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, Jan. 2009.

[44] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 164–174, Nov. 2010.

[45] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[46] J. M. Winn and C. M. Bishop, "Variational message passing," *J. Mach. Learning Research*, vol. 6, pp. 661–694, Apr. 2005.

[47] M. J. Beal, "Variational algorithms for approximate bayesian inference," Ph.D. dissertation, University College London, 2003.

[48] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[49] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[50] R. McEliece, D. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb 1998.

[51] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 219–230, Feb. 1998.

[52] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar 1974.

[53] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins Univ. Press, 1996.

[54] T. Heskes, "Stable fixed points of loopy belief propagation are local minima of the Bethe free energy," in *Advances in neural information processing systems*, 2002, pp. 343–350.

[55] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum–product algorithm," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4422–4437, Dec. 2007.

[56] Q. Su and Y.-C. Wu, "On convergence conditions of Gaussian belief propagation," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1144–1155, May 2015.

[57] M. K. Ng, *Iterative Methods for Toeplitz Systems*. Oxford University Press, 2004.

Paper B

Superfast Line Spectral Estimation

Thomas L. Hansen, Bernard H. Fleury and Bhaskar D. Rao

The paper has been accepted for publication in the
IEEE Transactions on Signal Processing

The layout has been revised.

Abstract

A number of recent works have proposed to solve the line spectral estimation problem by applying off-the-grid extensions of sparse estimation techniques. These methods are preferable over classical line spectral estimation algorithms because they inherently estimate the model order. However, they all have computation times which grow at least cubically in the problem size, thus limiting their practical applicability in cases with large dimensions. To alleviate this issue, we propose a low-complexity method for line spectral estimation, which also draws on ideas from sparse estimation. Our method is based on a Bayesian view of the problem. The signal covariance matrix is shown to have Toeplitz structure, allowing superfast Toeplitz inversion to be used. We demonstrate that our method achieves estimation accuracy at least as good as current methods and that it does so while being orders of magnitudes faster.

B.1 Introduction

The problem of line spectral estimation (LSE) has received significant attention in the research community for at least 40 years. The reason is that many fundamental problems in signal processing can be recast as LSE; examples include direction of arrival estimation using sensor arrays [1, 2], bearing and range estimation in synthetic aperture radar [3], channel estimation in wireless communications [4] and simulation of atomic systems in molecular dynamics [5].

In trying to solve the LSE problem, classical approaches include subspace methods [6] such as MUSIC [7] or ESPRIT [8] which estimate the frequencies based on an estimate of the signal covariance matrix. These approaches must be augmented with a method for estimation of the model order. Popular choices include generic information theoretic criteria (e.g. AIC, BIC) or more specialized methods, such as SORTE [9] which is based on the eigenvalues of the estimated signal covariance matrix. Subspace methods typically perform extremely well if the model order is known, but their estimation accuracy can degrade significantly if the model order is unknown.

The stochastic maximum likelihood (ML) method is known to be asymptotically efficient (it attains the Cramér-Rao bound as the problem size tends to infinity) [2]. Unfortunately it also requires knowledge of the model order.

Inspired by the ideas of sparse estimation and compressed sensing, many papers on sparsity-based LSE algorithms have appeared in recent years, e.g. [1, 10]. In particular, the LSE problem is simplified to a finite sparse reconstruction problem by restricting the frequencies to a grid. Such methods inherently estimate the model order, alleviating the issues arising from separate model order and frequency estimation in classical methods. The granularity of the grid leads to a non-trivial trade-off between accuracy and

computational requirements. To forego the use of a grid, so-called off-the-grid compressed sensing methods have been proposed [11–13]. These methods provably recover the frequencies in the noise-free case under a minimum separation condition. They suffer from prohibitively high computational requirements even for moderate problem sizes, see Sec B.6.

In [14–16] a Bayesian view is taken on the LSE problem. The model used in stochastic ML is extended with a sparsity-promoting prior on the coefficients of the sinusoid components. Thereby inherent estimation of the model order is achieved. These algorithms generally have high estimation accuracy. Their per-iteration computational complexity is cubic in the number of sinusoidal components, meaning that their runtime grows rapidly as the number of components increases.

In this work we introduce the Superfast LSE algorithm for solving the LSE problem in scenarios where the full measurement vector is available (complete data case). The modelling and design of the basic algorithm which we present in Sec. B.2 is based upon the ideas in [14–16]. The main novelty resides in the computational aspects of Superfast LSE. The derived method is based upon several techniques: a so-called superfast Toeplitz inversion algorithm [17, 18] (thereof the name of our algorithm), low-complexity Capon beamforming [19], the Gohberg-Semencul formula [20] and non-uniform fast Fourier transforms [21, 22]. The Superfast LSE algorithm has the following virtues: It inherently estimates all model parameters such as the noise variance and model order and it has low per-iteration computational complexity. Specifically it scales as $\mathcal{O}(N \log^2 N)$ where N is the length of the observed vector. We show empirically that it converges after a few iterations (typically less than 20). This means that for large problem sizes our algorithm can have computation time orders of magnitude lower than that of current methods. It does so without any penalty in estimation accuracy. Our numerical experiments show that Superfast LSE has high estimation accuracy across a wide range of scenarios, being on par with or better than state-of-the-art algorithms.

Synergistically and computationally efficiently combining the steps in the algorithm might appear easy after the fact. This is however not the case. Some other LSE algorithms can benefit in terms of computational effort from our approach, yet not to the extent achieved with the proposed algorithm. For instance, the computational methods in Sec. B.3 can be embedded in VALSE [16]. The resulting scheme will have high computational complexity due to the variational estimation of the posterior on the frequencies. Note that our algorithm performs on par with VALSE, but at a significantly reduced computational effort.

For completeness we also present a semifast version of the algorithm which works when only a subset of entries in the measurement vector are available. The Semifast LSE algorithm has per-iteration complexity $\mathcal{O}(N\hat{K}^2 +$

$N \log N)$, where $\hat{K}$ is the number of estimated sinusoids. Algorithms with similar per-iteration complexity are derived in [14, 15, 23, 24]. We have observed that our algorithm converges in a smaller number of iterations when compared to the algorithm in [15], thus leading to lower total runtime.

Outline: In Sec. B.2 we present our modelling and algorithm for LSE. Our low-complexity computational methods are presented in Sec. B.3 (complete date case) and B.4 (incomplete data case). In Sec. B.5 the algorithm is extended to the case of multiple measurement vectors. Numerical experiments are presented in Sec. B.6 and conclusions are given in Sec. B.7.

Notation: We write vectors as $\boldsymbol{a}$ and matrices as $\mathbf{A}$. The ith entry of vector $\boldsymbol{a}$ is denoted a_i or $[\boldsymbol{a}]_i$; the i,jth entry of matrix $\mathbf{A}$ is denoted $\mathbf{A}_{i,j}$. Let $\boldsymbol{b}$ be a binary vector (containing only zeros and ones) of the same dimension as $\boldsymbol{a}$, then $\boldsymbol{a}_{\boldsymbol{b}}$ denotes a vector which contains those entries in $\boldsymbol{a}$ where the corresponding entry in $\boldsymbol{b}$ is one. The Hadamard (entrywise) product is denoted by $\odot$.

B.2 An Algorithm for Line Spectral Estimation

We now detail the observation model and the specific objective of the LSE problem. The observation vector $\boldsymbol{y} \in \mathbb{C}^M$ contains time-domain samples and is given by

$$\boldsymbol{y} = \sum_{k=1}^{K} \boldsymbol{\Phi}\boldsymbol{\psi}(\tilde{\theta}_k)\tilde{\alpha}_k + \boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}} + \boldsymbol{w}, \tag{B.1}$$

where the steering vector function $\boldsymbol{\psi}(\theta_k) : [0,1) \to \mathbb{C}^{N\times 1}$ gives a Fourier vector, i.e., it has nth entry $[\boldsymbol{\psi}(\theta_k)]_n \triangleq \exp(j2\pi(n-1)\theta_k)$ for $n = 1,\dots,N$. We also define $\boldsymbol{\Psi}(\boldsymbol{\theta}) \triangleq [\boldsymbol{\psi}(\theta_1), \cdots, \boldsymbol{\psi}(\theta_{\dim(\boldsymbol{\theta})})]$. The measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$ is either the identity matrix ($M = N$, complete data case) or made of a subset of rows of a diagonal matrix ($M < N$, incomplete data case). The vector $\boldsymbol{w}$ is a white Gaussian noise vector with component variance β. The LSE problem is that of recovering the model order K along with the frequency $\tilde{\theta}_k \in [0,1]$ and coefficient $\tilde{\alpha}_k \in \mathbb{C}$ of each component $k = 1,\dots,K$.

B.2.1 Estimation Model

The estimation model and inference approach we present in the following are adaptions of ideas currently available in the literature. We have carefully combined these ideas to obtain an iterative scheme which can be implemented with low complexity as described in Secs. B.3 and B.4, while achieving a performance comparable to that of state-of-the-art algorithms.

Our algorithm is based on Bayesian inference in an estimation model which approximates (B.1). Specifically, to enable estimation of the model order K, we follow [14, 16] and employ a model with $K_{\max} \geq K$ components[1]. Each component has an associated activation variable $z_k \in \{0, 1\}$ which is set to 0 or 1 to deactivate or activate it. The activation variables are collected in the sparse vector $\boldsymbol{z}$. The effective estimated model order is given by the number of active components. Based on (B.1) we write our estimation model

$$\boldsymbol{y} = \sum_{k=1}^{K_{\max}} \boldsymbol{\Phi}\boldsymbol{\psi}(\theta_k) z_k \alpha_k + \boldsymbol{w} = \mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}} + \boldsymbol{w}, \tag{B.2}$$

where $\theta_k \in [0, 1)$ and $\alpha_k \in \mathbb{C}$ are frequencies and coefficients for $k = 1, \ldots, K_{\max}$ and we have defined $\mathbf{A}(\boldsymbol{\theta}) \triangleq \boldsymbol{\Phi}\boldsymbol{\Psi}(\boldsymbol{\theta})$.

Due to the Gaussian noise assumption we have

$$p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{z}, \boldsymbol{\theta}; \beta) = \mathrm{CN}(\boldsymbol{y}; \mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}}, \beta\mathbf{I}), \tag{B.3}$$

where $\mathrm{CN}(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the probability density function of a circularly symmetric complex normal random variable $\boldsymbol{y}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We assume $\beta \in [\varepsilon_\beta, \infty)$, where $\varepsilon_\beta > 0$ is an arbitrarily small constant which guarantees that the likelihood function is bounded below. A Bernoulli prior is used to promote deactivation of some of the components:

$$p(\boldsymbol{z}; \zeta) = \prod_{k=1}^{K_{\max}} \zeta^{z_k} (1 - \zeta)^{1 - z_k}, \tag{B.4}$$

where $\zeta \in [0, 1/2]$ is the activation probability. The restriction $\zeta \leq 1/2$ ensures that the prior is sparsity inducing. The coefficients are assumed to be independent zero-mean Gaussian

$$p(\boldsymbol{\alpha}; \boldsymbol{\gamma}) = \prod_{k=1}^{K_{\max}} \mathrm{CN}(\alpha_k; 0, \gamma_k), \tag{B.5}$$

where $\gamma_k \in [0, \infty)$ is the active-component variance. Sparsity-promoting priors have previously been used for both basis selection [25] and LSE [15]. The Bernoulli-Gaussian prior structure that we have adopted above was first introduced in [26] and used for LSE in [16].

Even though each α_k is modelled as Gaussian in (B.5), the prior specification is significantly more general than that because the variance of each component is estimated through γ_k. In the numerical investigation we demonstrate that our method works well even when the true density of each coefficient is not Gaussian.

[1]Since we can never expect to estimate more parameters than the number of observed observations, we select $K_{\max} = M$ in our implementation.

We finally use an independent and identically distributed (i.i.d.) uniform prior on the entries in $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}) = \prod_{k=1}^{K_{\max}} p(\theta_k) = \prod_{k=1}^{K_{\max}} 1 = 1. \tag{B.6}$$

If further prior information about the frequencies is available, it can easily be incorporated through $p(\boldsymbol{\theta})$.

B.2.2 Approach

By integrating the component coefficients we obtain the marginal likelihood

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\theta};\beta,\boldsymbol{\gamma}) &= \int p(\boldsymbol{y}|\boldsymbol{\alpha},\boldsymbol{z},\boldsymbol{\theta};\beta)p(\boldsymbol{\alpha};\boldsymbol{\gamma})\,\mathrm{d}\boldsymbol{\alpha} \\ &= \mathrm{CN}(\boldsymbol{y};\mathbf{0},\mathbf{C}) \end{aligned} \tag{B.7}$$

with $\mathbf{C} \triangleq \beta\mathbf{I} + \mathbf{A}(\boldsymbol{\theta}_z)\boldsymbol{\Gamma}_z\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}_z)$ and $\boldsymbol{\Gamma}_z \triangleq \mathrm{diag}(\boldsymbol{\gamma}_z)$.

Based on the marginal likelihood we can write the objective

$$\begin{aligned} \mathcal{L}(\boldsymbol{z},&\zeta,\beta,\boldsymbol{\theta},\boldsymbol{\gamma}) \triangleq -\ln p(\boldsymbol{z},\boldsymbol{\theta}|\boldsymbol{y};\beta,\boldsymbol{\gamma},\zeta) \\ &= -\ln p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{\theta};\beta,\boldsymbol{\gamma})p(\boldsymbol{z};\zeta)p(\boldsymbol{\theta}) + \text{const.} \\ &= \ln|\mathbf{C}| + \boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y} \\ &\quad - \sum_{k=1}^{K_{\max}} (z_k \ln\zeta + (1-z_k)\ln(1-\zeta)) + \text{const.} \end{aligned} \tag{B.8}$$

The variables $(\boldsymbol{z},\boldsymbol{\theta})$ and model parameters $(\beta,\boldsymbol{\gamma},\zeta)$ are estimated by minimizing (B.8), i.e., we seek the maximum a-posteriori (MAP) estimate of $(\boldsymbol{z},\boldsymbol{\theta})$ and the ML estimate of $(\beta,\boldsymbol{\gamma},\zeta)$. Our algorithm employs a block-coordinate descent method to find a local minimum (or saddle point) of (B.8).

For fixed $\boldsymbol{z}$ the first two terms in (B.8) are equal to the objective function of stochastic ML [2], and our approach can therefore be viewed as stochastic ML extended with a variable model order.

When the above estimates have been computed, the estimated model order is given by the number of active components, i.e. $\hat{K} = ||\hat{\boldsymbol{z}}||_0$, and the entries of $\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}$ are the estimated frequencies. The corresponding coefficients $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ can be estimated as follows. First, write the posterior of $\boldsymbol{\alpha}$ as

$$\begin{aligned} p(\boldsymbol{\alpha}|\boldsymbol{y},\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}};\hat{\beta},\hat{\boldsymbol{\gamma}}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha},\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}};\hat{\beta})p(\boldsymbol{\alpha};\hat{\boldsymbol{\gamma}}) \\ &\propto \mathrm{CN}(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}};\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}) \prod_{\{k:\hat{z}_k=0\}} \mathrm{CN}(\alpha_k;0,\hat{\gamma}_k), \end{aligned} \tag{B.9}$$

where

$$\hat{\boldsymbol{\mu}} \triangleq \hat{\beta}^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\boldsymbol{y} \tag{B.10}$$

$$\hat{\boldsymbol{\Sigma}} \triangleq \left(\hat{\beta}^{-1}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) + \hat{\boldsymbol{\Gamma}}_{\hat{\boldsymbol{z}}}^{-1}\right)^{-1}. \tag{B.11}$$

As expected the posterior of the coefficients corresponding to inactive components (those for which $\hat{z}_k = 0$) coincides with their prior. These are not of interest (they are inconsequential in the model (B.2)) and integrating them out gives a Gaussian posterior over $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$. If a point estimate of $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ is needed, the MAP (which is also the LMMSE) estimate $\hat{\boldsymbol{\alpha}}_{\hat{\boldsymbol{z}}} = \hat{\boldsymbol{\mu}}$ can be used[2].

B.2.3 Derivation of Update Equations

As mentioned, our algorithm is derived as a block-coordinate descent method applied on $\mathcal{L}$ in (B.8). The estimates are updated in the following blocks: $\hat{\boldsymbol{z}}$, $\hat{\boldsymbol{\zeta}}$, $\hat{\beta}$ and $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}, \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}})$. Each update is guaranteed not to increase $\mathcal{L}$. We note that the frequencies and variances of inactive components (those for which $\hat{z}_k = 0$) are not updated, as $\mathcal{L}$ does not depend on these variables.

Estimation of frequencies and coefficient variances

Even when all remaining variables are kept fixed, it is not tractable to find the global minimizer of $\mathcal{L}$ with respect to the vector of active component frequencies $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and variances $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$. We therefore resort to a numerical method. Writing only the terms of (B.8) which depend on $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$, we have

$$\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}) = \ln|\mathbf{C}| + \boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y} + \text{const.},$$

so we need to solve $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}, \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}) = \underset{(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})}{\arg\min}\, \mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$.

In [10] a similar optimization problem involving only the frequencies is solved by Newton's method. Directly applying that approach in our case leads to high computational complexity. Methods based on gradient descent have also been proposed [14], but we have observed that using this approach leads to slow converge. As we are concerned with computational speed in this paper, we instead use the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [27]. This algorithm only requires evaluation of the objective function and its gradient. In the following we demonstrate how these evaluations can be performed with low complexity. At the same time the per-iteration of L-BFGS is linear in $\hat{K}$, namely $\mathcal{O}(J\hat{K})$, where J is the number of saved updates used in L-BFGS. In our implementation we

[2]Note that for computational convenience we write $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}} \odot \boldsymbol{q}$, where $\boldsymbol{q}$ is defined by (B.16). See the text following (B.26).

use $J = 10$. We have observed that L-BFGS converges in a small number of iterations.

The L-BFGS algorithm requires an initial estimate of the Hessian of $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$, which is subsequently updated in each iteration of the algorithm. Every update of the activation variable $\hat{\boldsymbol{z}}$ results in a change in the dimension of the Hessian (the number of variables in $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ changes). This means that the implicit estimate of the Hessian in the L-BFGS algorithm is reinitialized rather frequently in our estimation scheme. As a result, the degree of accuracy of the initialization of the Hessian has a significant impact on the convergence speed of the algorithm. We therefore propose to initialize L-BFGS with a diagonal approximation of the Hessian. As shown below, the diagonal entries of the Hessian can be obtained with low computational complexity.

The initial estimate of the Hessian must be positive definite. This is only achieved when all diagonal entries are positive. Those entries of the diagonal Hessian which are negative are therefore replaced with the following values: For entries corresponding to frequency variables we use $(50N)^2$ as the diagonal Hessian and for the entries corresponding to the variance of the kth component we use $[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k^{-2}$. These heuristic values have been determined by considering a diagonally scaled version of the optimization problem (see [28, Sec. 1.3]).

Here follows the required first- and second-order partial derivatives of $\mathcal{L}(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ evaluated at the current estimates $(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}, \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}})$ (see [15] for some hints on how these are obtained):

$$\frac{\partial \mathcal{L}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k} = 2[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k \operatorname{Im}\{t_k - q_k^* r_k\} \tag{B.12}$$

$$\frac{\partial \mathcal{L}}{\partial [\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}]_k} = s_k - |q_k|^2 \tag{B.13}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial [\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}]_k^2} = 2[\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k \operatorname{Re}\big\{ & x_k - v_k + [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k (t_k^2 - x_k s_k) \\ & + [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k (x_k |q_k|^2 + s_k |r_k|^2 - 2 t_k r_k q_k^*) + (u_k q_k^* - |r_k|^2) \big\}\end{aligned} \tag{B.14}$$

$$\frac{\partial^2 \mathcal{L}}{\partial [\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}]_k^2} = 2 s_k |q_k|^2 - s_k^2, \tag{B.15}$$

where we have defined vectors

$$\boldsymbol{q} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{y} \tag{B.16}$$

$$\boldsymbol{r} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{y} \tag{B.17}$$

$$\boldsymbol{s} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big) \tag{B.18}$$

$$\boldsymbol{t} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big) \tag{B.19}$$

$$\boldsymbol{u} \triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}^2\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{y} \tag{B.20}$$

$$\boldsymbol{v} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}^2\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big) \tag{B.21}$$

$$\boldsymbol{x} \triangleq \mathrm{diag}\Big(\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{D}\boldsymbol{\Phi}^{\mathrm{H}}\hat{\mathbf{C}}^{-1}\boldsymbol{\Phi}\mathbf{D}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\Big)\,. \tag{B.22}$$

The notation $\mathrm{diag}(\cdot)$ denotes a vector composed of the diagonal entries of the (matrix) argument. The matrix $\hat{\mathbf{C}}$ is that in (B.7) evaluated at $\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}$, $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$ and $\hat{\beta}$. We have defined the diagonal matrix $\mathbf{D} \triangleq \mathrm{diag}\big([0, 2\pi, 4\pi, \ldots, (N-1)2\pi]^{\mathrm{T}}\big)$. In Secs. B.3 and B.4 we discuss how the vectors (B.16)–(B.22) can be calculated with low computational complexity.

Estimation of activation probability

With all other variables fixed, the objective (B.8) is a convex function of $\zeta \in [0, 1/2]$ $\left(\frac{\partial^2 \mathcal{L}}{\partial \zeta^2} > 0\right)$. The global minimizer is then found by differentiating and setting equal to zero. Considering the constraints on ζ, we update it as

$$\hat{\zeta} = \min\left(\frac{1}{2}, \frac{||\hat{\boldsymbol{z}}||_0}{K_{\max}}\right). \tag{B.23}$$

Estimation of noise variance

Even when keeping all remaining variables fixed at their current estimate, the globally minimizing noise variance β in (B.8) cannot be found in closed form. An obvious alternative approach would be to incorporate the estimation of β into L-BFGS together with the estimation of $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$. However, we have observed this approach to exhibit slow convergence because the objective function can be rather "flat" in the variable β (the gradient is small far away from any stationary point).

In sparse Bayesian learning [25] a similar estimation problem is solved successfully via the expectation-minimization (EM) algorithm. To use EM, we need to reintroduce $\boldsymbol{\alpha}$ into the estimation problem. In order to show how EM is integrated into our coordinate-block descent method and that the update of $\hat{\beta}$ is guaranteed not to increase (B.8), it is the easiest to directly use the upper bound associated with EM (see [29] for a derivation of EM which takes a similar approach).

The updated estimate of β is the minimizer of an upper bound on the objective function (B.8). To obtain the upper bound we write the terms of the objective function which depend on β, with all other variables kept fixed at their current estimates:

$$\begin{aligned}\mathcal{L}(\beta) &= -\ln p(\boldsymbol{y}|\hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}}) + \text{const.} \\ &= -\ln \int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \frac{p(\boldsymbol{y}, \boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}|\hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}})}{f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})} \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} + \text{const.} \\ &\leq -\int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}|\hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \beta, \hat{\boldsymbol{\gamma}})}{f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})} \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} + \text{const.}, \end{aligned} \tag{B.24}$$

where $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \geq 0$ is a function which fulfills $\int f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) \, \mathrm{d}\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}} = 1$. The inequality follows from Jensen's inequality.

Following EM, we select $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}) = p(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}|\boldsymbol{y}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}}; \hat{\beta}^{i-1}, \hat{\boldsymbol{\gamma}})$, where $\hat{\beta}^{i-1}$ denotes the previous noise variance estimate. Denote the upper bound on the right-hand side of (B.24) by $Q(\beta; \hat{\beta}^{i-1})$ and insert $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})$ to get

$$\begin{aligned}Q(\beta; \hat{\beta}^{i-1}) = M \ln \beta + \beta^{-1} \operatorname{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\right) \\ + \beta^{-1} ||\boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \hat{\boldsymbol{\mu}}||^2 + \text{const.}, \end{aligned} \tag{B.25}$$

where we have used (B.9) to evaluate expectations involving $\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}$ and $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are calculated from (B.10)–(B.11) based on $\hat{\beta}^{i-1}$. It is easy to show that the upper bound has a unique minimizer, which is used as the updated estimate of the noise variance:

$$\hat{\beta}^i = \max\left(\varepsilon_\beta, \frac{\operatorname{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\right) + ||\boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \hat{\boldsymbol{\mu}}||^2}{M}\right). \tag{B.26}$$

To allow low-complexity calculation of $\hat{\beta}^i$ we use Woodbury's matrix inversion identity to show that $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}} \odot \boldsymbol{q}$ and $\operatorname{tr}\left(\hat{\boldsymbol{\Sigma}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}) \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\right) = \sum_{k=1}^{\hat{K}} (\hat{\beta}^{i-1} s_k [\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}]_k)$.

The update (B.26) could be applied repeatedly since an improved upper bound is used each time. Since we have not observed any advantages by doing so, we simply perform the update (B.26) once for each pass in the block-coordinate descent algorithm. We also note that even though EM is known to be prone to slow convergence speed, we have observed empirically that the estimate of β converges fast, typically within 10 iterations.

It can easily be shown that with the chosen $f(\boldsymbol{\alpha}_{\hat{\boldsymbol{z}}})$, the inequality in (B.24) holds with equality at $\beta = \hat{\beta}^{i-1}$. It then follows that the new estimate of β does not increase the value of the objective function (see the proof of Lemma B.4 in Appendix A).

Deactivation of Components

We now describe the activation and deactivation of components, which is performed by the single most likely replacement (SMLR) detector [26]. SMLR has previously been demonstrated to perform well for LSE [14–16, 23].

First we write the terms of (B.8) which depend on the variables pertaining to the kth component and fix all other variables at their current estimate. Based on Woodbury's matrix inversion identity and the determinant lemma we get (see [24] for details)

$$\mathcal{L}(z_k, \theta_k, \gamma_k) = z_k \Bigg(- \frac{|q_{\sim k}(\theta_k)|^2}{\gamma_k^{-1} + s_{\sim k}(\theta_k)} + \ln\Bigg(\left(1 + \gamma_k s_{\sim k}(\theta_k)\right) \frac{1-\hat{\zeta}}{\hat{\zeta}} \Bigg) \Bigg) + \text{const.}, \quad \text{(B.27)}$$

with

$$\begin{aligned} q_{\sim k}(\theta_k) &\triangleq \boldsymbol{\psi}^{\mathrm{H}}(\theta_k) \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}_{\sim k}^{-1} \boldsymbol{y} \\ s_{\sim k}(\theta_k) &\triangleq \boldsymbol{\psi}^{\mathrm{H}}(\theta_k) \boldsymbol{\Phi}^{\mathrm{H}} \hat{\mathbf{C}}_{\sim k}^{-1} \boldsymbol{\Phi} \boldsymbol{\psi}(\theta_k), \end{aligned} \quad \text{(B.28)}$$

where $\hat{\mathbf{C}}_{\sim k} \triangleq \hat{\beta}\mathbf{I} + \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}_{\sim k}}) \hat{\boldsymbol{\Gamma}}_{\hat{\boldsymbol{z}}_{\sim k}} \mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}_{\sim k}})$ and $\hat{\boldsymbol{z}}_{\sim k}$ is equal to $\hat{\boldsymbol{z}}$ with the kth entry forced to zero. The matrix $\hat{\mathbf{C}}_{\sim k}$ is thus the marginal covariance matrix of the observation vector with the kth component deactivated.

To evaluate if an active component should be deactivated, we test if the objective $\mathcal{L}$ is increased by doing so, i.e., we test if $\mathcal{L}(z_k = 0, \hat{\theta}_k, \hat{\gamma}_k) < \mathcal{L}(z_k = 1, \hat{\theta}_k, \hat{\gamma}_k)$. This gives the deactivation criterion for the kth component:

$$\frac{|q_{\sim k}(\hat{\theta}_k)|^2}{\hat{\gamma}_k^{-1} + s_{\sim k}(\hat{\theta}_k)} - \ln\big(1 + \hat{\gamma}_k s_{\sim k}(\hat{\theta}_k)\big) < \ln\left(\frac{1-\hat{\zeta}}{\hat{\zeta}}\right). \quad \text{(B.29)}$$

This criterion is evaluated for currently active components, i.e., for k which has corresponding $\hat{z}_k = 1$.

For computational convenience we note that we can obtain $q_{\sim k}(\hat{\theta}_k)$ and $s_{\sim k}(\hat{\theta}_k)$ from $\boldsymbol{q}$ and $\boldsymbol{s}$ with low complexity. First, write

$$\hat{\mathbf{C}}_{\sim k} = \hat{\mathbf{C}} - \hat{\gamma}_k \boldsymbol{\Phi} \boldsymbol{\psi}(\hat{\theta}_k) \boldsymbol{\psi}^{\mathrm{H}}(\hat{\theta}_k) \boldsymbol{\Phi}^{\mathrm{H}}$$

and use Woodbury's identity to obtain

$$\begin{aligned} q_{\sim k}(\hat{\theta}_k) &= \frac{q_i}{1 - \hat{\gamma}_k s_i} \\ s_{\sim k}(\hat{\theta}_k) &= \frac{s_i}{1 - \hat{\gamma}_k s_i}, \end{aligned}$$

where q_i and s_i are the ith entries of (B.16) and (B.18) with i denoting the index for which $[\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}}]_i = \hat{\theta}_k$.

Component Activation

We now describe a method to decide if a deactivated component should be activated. This also involves estimating the frequency and variance of this component, because no meaningful such estimates are available before the component is activated. Any of the deactivated components are equally good candidates for activation. In the following k refers to an arbitrary value for which $\hat{z}_k = 0$. If no such k exists all components are already activated and the activation step is not carried out.

Our method is again based on the expression (B.27). Inspired by [16], let $\bar{\gamma}$ denote the average of the entries in $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$. Define the change in the objective obtained from setting $\hat{z}_k = 1$, $\hat{\theta}_k = \theta_k$, $\hat{\gamma}_k = \bar{\gamma}$:

$$\begin{aligned}\Delta\mathcal{L}(\theta_k) &= \mathcal{L}(1, \theta_k, \bar{\gamma}) - \mathcal{L}(0, \theta_k, \bar{\gamma}) \\ &= \ln\left((1 + \bar{\gamma} s_{\sim k}(\theta_k))\, \frac{1-\hat{\zeta}}{\hat{\zeta}}\right) - \frac{|q_{\sim k}(\theta_k)|^2}{\bar{\gamma}^{-1} + s_{\sim k}(\theta_k)}\end{aligned} \tag{B.30}$$

Note that the last term in (B.30) does not depend on θ_k or $\bar{\gamma}$. Then the frequency is found by maximizing the decrease in the objective, i.e.,

$$\hat{\theta}_k = \underset{\theta_k \in \mathcal{G}}{\arg\min}\, \Delta\mathcal{L}(\theta_k), \tag{B.31}$$

where $\mathcal{G}$ is a grid of L equispaced values, i.e., $\mathcal{G} \triangleq \{0, 1/L, \ldots, 1 - 1/L\}$. The restriction of the estimated frequencies to a grid does not mean that the final frequency estimates lie on a grid, because they are refined to be in $[0, 1)$ in subsequent updates of the frequency vector. For this reason, the choice of L does not have any impact on the estimation accuracy, provided that it is sufficiently large.[3] In Sec. B.3 and B.4 we show how $q_{\sim k}(\theta_k)$ and $s_{\sim k}(\theta_k)$ can be evaluated with low complexity for all $\theta_k \in \mathcal{G}$, such that the minimization can be performed by means of an exhaustive search over $\mathcal{G}$.

The activation procedure continues only if a decrease in the objective can be obtained by activating a component at $\hat{\theta}_k$, i.e., if $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$. The inclusion of the constant $\varepsilon_{\mathcal{L}} > 0$ is purely technical, as it simplifies our convergence analysis. It can be chosen arbitrarily small and we select it as machine precision in our implementation.

After estimating the frequency, the component variance is selected as $\hat{\gamma}_k = \underset{\gamma_k}{\arg\min}\, \Delta\mathcal{L}(1, \hat{\theta}_k, \gamma_k)$. Using an approach similar to [24], this minimizer can

[3] A numerical investigation (not reported here) shows that the algorithm is invariant to the choice of L, provided that $L \geq 2N$. In our implementation we use L equal to $8N$ rounded to the nearest power of 2.

be shown to be

$$\hat{\gamma}_k = \begin{cases} \frac{|q_{\sim k}(\theta_k)|^2 - s_{\sim k}(\theta_k)}{s^2_{\sim k}(\theta_k)} & \text{if } \frac{|q_{\sim k}(\theta_k)|^2}{s_{\sim k}(\theta_k)} > 1, \\ 0 & \text{otherwise.} \end{cases} \tag{B.32}$$

The component is only activated if[4] $\hat{\gamma}_k > 0$.

It is instructive to explore the activation criterion $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$ in detail. Since $\varepsilon_{\mathcal{L}}$ is machine precision, we ignore it ($\varepsilon_{\mathcal{L}} = 0$) for simplicity. The activation criterion can be rewritten to the form

$$\frac{|q_{\sim k}(\hat{\theta}_k)|^2}{s_{\sim k}(\hat{\theta}_k)} > \left(1 + \frac{1}{\bar{\gamma} s_{\sim k}(\hat{\theta}_k)}\right) \ln\left(\left(1 + \bar{\gamma} s_{\sim k}(\hat{\theta}_k)\right) \frac{1 - \hat{\zeta}}{\hat{\zeta}}\right). \tag{B.33}$$

Denote the left-hand side of (B.33) as κ_k. This quantity can be interpreted as the signal-to-noise ratio of the kth component [30, 31]. If the sparse Bayesian learning (SBL) model is used for sparsity promotion, an activation criterion of the from $\kappa_k > 1$ is obtained [30, 31]. Algorithms using the activation criterion $\kappa_k > 1$ are known to be prone to the activation of "artefact" components with very small $\hat{\gamma}_k$ and $\hat{\alpha}_k$ at what seems to be arbitrary frequencies $\hat{\theta}_k$. The right-hand side of (B.33) is always larger than one and this helps reduce the number of artefacts which are activated, as demonstrated in [16]. This favorable phenomenon is caused by the use of the average $\bar{\gamma}$ in the definition of $\Delta\mathcal{L}(\theta_k)$ (as opposed to inserting $\hat{\gamma}_k$ from (B.32), which resembles the SBL approach).

Even still, we have observed the activation of a few artefact components in our numerical investigations. We therefore follow the same idea as [30, 31] and heuristically adjust the criterion (B.33) to obtain

$$\frac{|q_{\sim k}(\theta_k)|^2}{s_{\sim k}(\theta_k)} > \left(1 + \frac{1}{\bar{\gamma} s_{\sim k}(\theta_k)}\right) \ln\left(\left(1 + \bar{\gamma} s_{\sim k}(\theta_k)\right) \frac{1 - \hat{\zeta}}{\hat{\zeta}}\right) + \tau, \tag{B.34}$$

where $\tau \geq 0$ is some adjustment of the threshold. Specifically we select $\tau = 5$, cf. the numerical study in Sec. B.6.2. Our numerical experiments show that this simple approach is very effective at avoiding the inclusion of small spurious components. Since the heuristic criterion (B.34) is stricter than the criterion $\Delta\mathcal{L}(\hat{\theta}_k) < -\varepsilon_{\mathcal{L}}$, it is guaranteed that the activation of a component decreases the objective function.

B.2.4 Outline of the Algorithm and Implementation Details

The algorithm proceeds by repeating the following steps until convergence:

[4] When $\hat{\gamma}_k = 0$ the kth component is effectively deactivated because the corresponding coefficient α_k has a zero-mean prior with zero variance, see (B.5). The effective deactivation is also seen in the definition of $\mathbf{C}$ in (B.7) and it further manifests itself as $\hat{\mu}_k = 0$ in (B.10).

1. Check if any components can be activated via the procedure described in Sec. B.2.3.
2. Re-estimate the activation probability ζ via (B.23).
3. Re-estimate the noise variance via (B.26).
4. Repeat:
   4a) Perform a single L-BFGS update of the estimated vectors of active component frequencies $\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}$ and variances $\boldsymbol{\gamma}_{\hat{\boldsymbol{z}}}$, as described in Sec. B.2.3.
   4b) Check if any components can be deactivated via (B.29).

The algorithm terminates when the change in the objective (B.8) between two consecutive iterations is less than $M10^{-7}$.

In step 1) and 4b) the check for component (de)activation is repeated until no more components can be (de)activated. The updates in step 4) are iterated until either the approximated squared Newton decrement of the L-BFGS method is below $M10^{-8}$ or at most 5 times.

The observant reader will have noticed that the minimization over $(\boldsymbol{\theta}_{\hat{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\hat{\boldsymbol{z}}})$ must be constrained to $\gamma_k \geq 0$ for all k. It turns out that this constraint can be handled in a simple manner: Notice that the deactivation criterion (B.29) is always fulfilled for $\hat{\gamma}_k$ sufficiently small. The constraint is therefore never active at the solution. We therefore simply need to restrict the line-search performed in L-BFGS such that the no entry in $\hat{\boldsymbol{\gamma}}_{\hat{\boldsymbol{z}}}$ ever becomes negative. If any $\hat{\gamma}_k$ approaches (or becomes equal to) zero, it is deactivated in step 4b). Note that this approach resembles that of L-BFGS for box constraints [32], except that the deactivation of variables for which the constraint is active happens automatically in our algorithm.

The algorithm is initialized with all components in the deactivated stage (i.e. $\hat{\boldsymbol{z}} = \boldsymbol{0}$). The initial values of the entries in $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\gamma}}$ do not matter, since they are assigned when their corresponding component is activated (see Sec. B.2.3). The noise variance is initialized to $\hat{\beta} = 0.01||\boldsymbol{y}||^2/M$ (1 % of the energy in $\boldsymbol{y}$ is assumed to be noise). The activation probability is initialized to $\hat{\zeta} = 0.2$.

In Appendix A we discuss in detail the convergence properties of our algorithm. The findings are summarized here. We show that our algorithm terminates in finite time and that the estimates of $\boldsymbol{z}$, ζ and β are guaranteed to converge. We denote the limit points as $\bar{\boldsymbol{z}}$, $\bar{\zeta}$ and $\bar{\beta}$. When these estimates have converged, our algorithm reduces to a pure L-BFGS scheme which estimates $(\boldsymbol{\theta}_{\bar{\boldsymbol{z}}}, \boldsymbol{\gamma}_{\bar{\boldsymbol{z}}})$. Due to the non-convexity of the objective function, we cannot guarantee convergence of L-BFGS (see [33]). Despite of this, we have never observed non-convergence of our algorithm. In our experiments it always

converged to a local minimum of the objective function. We therefore rely on the vast amount of experimental validation of the convergence of L-BFGS and assume convergence to a stationary point. In particular, we have the following theorem.

Theorem B.1. *Assume that L-BFGS in step 4a) converges to a stationary point of* $(\boldsymbol{\theta}_{\bar{z}}, \boldsymbol{\gamma}_{\bar{z}}) \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \bar{\zeta}, \bar{\beta}, \boldsymbol{\theta}_{\bar{z}}, \boldsymbol{\gamma}_{\bar{z}})$*. Then the sequence of estimates obtained by our algorithm converges. Further, the limit point is a stationary point of* $(\zeta, \beta, \boldsymbol{\theta}, \boldsymbol{\gamma}) \mapsto \mathcal{L}(\bar{\boldsymbol{z}}, \zeta, \beta, \boldsymbol{\theta}, \boldsymbol{\gamma})$*, in the sense that the Karush-Kuhn-Tucker necessary conditions for a minimum are fulfilled.*

Proof: See Appendix A. ■

B.2.5 Initial Activation of Components

When the number of sinusoids K in the observed signal (B.1) is high, the algorithm spends significant computational effort activating components (step 1). This is because each time a component is activated, the values $q_{\sim k}(\theta_k)$ and $s_{\sim k}(\theta_k)$ must be evaluated for all $\theta_k \in \mathcal{G}$ to calculate (B.31). To alleviate the computational effort of building the initial set of active components, we propose to let the first few iterations use an approximate scheme for activating components in place of step 1). The approximate activation scheme proceeds as follows:

1. Calculate $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$, where k is the index of a deactivated component.
2. Evaluate $\Delta\mathcal{L}(\theta)$ (B.30) for all $\theta \in \mathcal{G}$.
3. Find the local minimizers of $\Delta\mathcal{L}(\theta)$, i.e., find the values of θ for which $\Delta\mathcal{L}(\theta) \leq \Delta\mathcal{L}(\theta')$ with θ' being any of the two neighbouring grid-points of θ. The local minimizers are candidate frequencies.
4. Activate a component at those candidate frequencies for which the following criteria are fulfilled:
   - The component activation criterion (B.33) is fulfilled.
   - The component variance (B.32) is non-zero.
   - The decrease in the objective obeys $\Delta\mathcal{L}(\theta) \leq \Delta\mathcal{L}_{\min}/5$, where $\Delta\mathcal{L}_{\min}$ is the largest decrease obtained from activating a component at another candidate frequency (in the current iteration).
   - All other currently active components have frequency estimates located at least[5] $0.05N^{-1}$ apart from the candidate frequency.

[5]For the distance measure we use the wrap-around distance on $[0,1)$ defined as $d(x,y) \triangleq \min(|x-y|, 1-|x-y|)$ for $x, y \in [0,1)$.

The above method is a heuristic scheme, which quickly builds a set of activated components. Typically this set is close to the final result and only a few (in our setup less than 15 in most cases) iterations are need before convergence.

B.3 Superfast Method (Complete Observations)

The algorithm presented above has rather large computational complexity, in particular due to the inversion of $\mathbf{C}$ and the calculation of $q_{\sim k}(\theta)$ $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$. In this section we discuss how all updates of the algorithm can be evaluated with low computational complexity by exploiting the inherent structure of the problem. In particular we discuss how to evaluate $\ln|\mathbf{C}|$, $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$, $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{x}$ and $q_{\sim k}(\theta)$, $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$.

The method presented here is only applicable when the complete observation vector is available, i.e., when $\boldsymbol{\Phi} = \mathbf{I}$, $M = N$ and $\mathbf{A}(\boldsymbol{\theta}) = \boldsymbol{\Psi}(\boldsymbol{\theta})$. In this case the observation vector $\boldsymbol{y}$ is a wide-sense stationary process and its covariance matrix $\mathbf{C}$ is Hermitian Toeplitz. Low-complexity algorithms for inverting such matrices are available in the literature. We also rely on fast Fourier transform (FFT) techniques.

Our approach is based on the Gohberg-Semencul formula [17, 20], which states that the inverse of the Hermitian Toeplitz matrix $\mathbf{C}$ can be decomposed as

$$\mathbf{C}^{-1} = \delta_{N-1}^{-1}\left(\mathbf{T}_1^{\mathrm{H}}\mathbf{T}_1 - \mathbf{T}_0\mathbf{T}_0^{\mathrm{H}}\right), \tag{B.35}$$

where the entries of $\mathbf{T}_0$ and $\mathbf{T}_1$ are

$$[\mathbf{T}_0]_{i,k} = \rho_{i-k-1},$$
$$[\mathbf{T}_1]_{i,k} = \rho_{N-1+i-k}$$

for $i,k = 1,\ldots,N$. Note that $\rho_i = 0$ for $i < 0$ and $i > N-1$; thus $\mathbf{T}_0$ is strictly lower triangular and $\mathbf{T}_1$ is unit upper triangular ($\rho_{N-1} = 1$). The values δ_i and ρ_i for $i = 0,\ldots,N-1$ can be computed with a generalized Schur algorithm in time $\mathcal{O}(N\log^2 N)$ [17]. Alternatively, the Levinson-Durbin algorithm can also be used to obtain the decomposition in time $\mathcal{O}(N^2)$. The latter algorithm is significantly simpler to implement and is faster for small N. In [18] it is concluded that the Levinson-Durbin algorithm requires fewer total operations than the generalized Schur algorithm for $N \leq 256$.

B.3.1 Evaluating $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ and $\ln|\mathbf{C}|$

To calculate the value of the objective function (B.8) we need to find $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ and $\ln|\mathbf{C}|$. Inspecting (B.35) it is clear that matrix-vector products involving

$\mathbf{T}_0$ and $\mathbf{T}_1$ are convolutions. These can be implemented using FFT techniques. The product $\boldsymbol{y}^{\mathrm{H}}\mathbf{C}^{-1}\boldsymbol{y}$ can thus be calculated in $\mathcal{O}(N \log N)$ time when $\{\rho_i\}$ and δ_{N-1} are known.

The matrix $\mathbf{C}$ is Hermitian positive definite and can therefore be factorized uniquely as

$$\mathbf{C} = \mathbf{L}\mathbf{B}\mathbf{L}^{\mathrm{H}}, \tag{B.36}$$

with $\mathbf{L}$ being unit lower triangular. The diagonal matrix $\mathbf{B}$ is computed with the generalized Schur algorithm. Its diagonal entries are given by δ_i for $i = 0, \ldots, N-1$ [17]. Since the determinant of a triangular matrix is the product of its diagonal entries, we have

$$\ln |\mathbf{C}| = \sum_{i=0}^{N-1} \ln \delta_i. \tag{B.37}$$

It follows that once the generalized Schur algorithm has been executed, the objective function (B.8) can easily be found.

B.3.2 Evaluating q, r and u

Note that $\mathbf{C}^{-1}\boldsymbol{y}$ can be evaluated with FFT techniques using (B.35). We recognize that matrix-vector products involving $\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ are Fourier transforms evaluated off the equispaced grid. Such products are approximated to a very high precision in time $\mathcal{O}(N \log N)$ using the non-uniform fast Fourier transform[6] (NUFFT) [21, 22]. Then $\boldsymbol{q}$, $\boldsymbol{r}$ and $\boldsymbol{u}$ are easily found in time $\mathcal{O}(N \log N)$ (assuming the decomposition (B.35) has already been calculated).

B.3.3 Evaluating s, t, v and x

Turning our attention to $\boldsymbol{s}$, we follow [19] and note that (recall that we assume $\boldsymbol{\Phi} = \mathbf{I}$)

$$\begin{aligned} s_k &= [\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{C}^{-1}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{z}})]_{k,k} \\ &= \sum_{i=-(N-1)}^{N-1} \omega_s(i) \exp\left(j2\pi i[\hat{\boldsymbol{\theta}}_{\hat{z}}]_k\right) \end{aligned} \tag{B.38}$$

[6] The NUFFT calculates the Fourier transform at arbitrary points (not lying on an equispaced grid) by interpolation combined with an FFT. It is an approximation, which can be made arbitrarily accurate by including more points in the interpolation. The NUFFT achieves a time complexity of $\mathcal{O}(N \log N + K)$, where K is the number of off-the-grid frequency points at which it is evaluated. For $K \leq N$ this complexity is equal to that of the FFT, but the constant hidden in the big-O notation is much higher for the NUFFT. We have found that for $N \geq 512$ significant speedups can be achieved by using the NUFFT over a direct computation of $\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ and evaluation of the matrix-vector products involving this matrix. In particular the speedup arises from the fact that $\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ no longer needs to be formed.

for $k = 1, \ldots, \hat{K}$ where $\hat{K}$ is the number of entries in $\hat{\boldsymbol{\theta}}_{\hat{z}}$. The function $\omega_s(i)$ gives the sum over the ith diagonal, i.e.,

$$\omega_s(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)} [\mathbf{C}^{-1}]_{q+1,q+i+1}. \tag{B.39}$$

It is obvious that (B.38) can be calculated for all $k = 1, \ldots, \hat{K}$ via a NUFFT when the values $\omega_s(i)$ are available.

To evaluate $\boldsymbol{t}$, $\boldsymbol{v}$ and $\boldsymbol{x}$ we follow a similar approach and note that the entries of these vectors can be written as (B.38) with $\omega_s(i)$ replaced by

$$\omega_t(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)} [\mathbf{D}\mathbf{C}^{-1}]_{q+1,q+i+1} \tag{B.40}$$

$$\omega_v(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)} [\mathbf{D}^2\mathbf{C}^{-1}]_{q+1,q+i+1} \tag{B.41}$$

$$\omega_x(i) = \sum_{q=\max(0,-i)}^{\min(N-1-i,N-1)} [\mathbf{D}\mathbf{C}^{-1}\mathbf{D}]_{q+1,q+i+1}, \tag{B.42}$$

respectively. In Appendix B we demonstrate how $\{\omega_s(i)\}$, $\{\omega_t(i)\}$, $\{\omega_v(i)\}$ and $\{\omega_x(i)\}$ can be obtained through length-$2N$ FFTs using the decomposition (B.35).

B.3.4 Evaluating $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta \in \mathcal{G}$

To calculate the frequency of the component processed in the activation stage, $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ must be evaluated for all $\theta \in \mathcal{G}$, where $\mathcal{G}$ is a grid of L equispaced points. Defining the vector of gridded frequencies $\boldsymbol{\theta}^{\mathcal{G}} \triangleq [0, 1/L, \ldots, (L-1)/L]^{\mathrm{T}}$, we need to find

$$\begin{aligned}
\boldsymbol{q}^{\mathcal{G}} &\triangleq \boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{C}^{-1}\boldsymbol{y}, \\
\boldsymbol{s}^{\mathcal{G}} &\triangleq \operatorname{diag}\left(\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{C}^{-1}\boldsymbol{\Psi}(\boldsymbol{\theta}^{\mathcal{G}})\right).
\end{aligned}$$

We have used the fact that in the beginning of the activation step the kth component is deactivated and thus $\mathbf{C}_{\sim k} = \mathbf{C}$.

Since $\mathcal{G}$ is an equispaced grid, products with $\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})$ can be evaluated as a length-L FFT. The vector $\boldsymbol{q}^{\mathcal{G}}$ is therefore easy to find. Rewriting $\boldsymbol{s}^{\mathcal{G}}$ in the form (B.38), it is seen that $\boldsymbol{s}^{\mathcal{G}}$ can also be evaluated as a length-L FFT. These computations have time-complexity $\mathcal{O}(L \log L)$ (assuming the decomposition (B.35) has already been calculated).

B.3.5 Algorithm Complexity

In summary, the time complexity of each iteration in the algorithm described in Sec. B.2 is dominated by either the calculation of $\{\rho_i\}$ and δ_{N-1} with the generalized Schur algorithm or the calculation of $\boldsymbol{q}^{\mathcal{G}}$ and $\boldsymbol{s}^{\mathcal{G}}$ (we assume $\hat{K} \leq M = N \leq L$). With our choice $L = 8N$ we have complexity per iteration of $\mathcal{O}(N \log^2 N)$.

Also note that all computations involving $\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ are performed using the NUFFT. This matrix therefore does not need to be stored, so our algorithm only uses a modest amount of memory.

B.4 Semifast Method (Incomplete Observations)

The method presented in Sec. B.3 is not applicable when an incomplete observation vector is available, i.e., when $\boldsymbol{\Phi} \neq \mathbf{I}$. In the following we introduce a computational method, which can be used when $\boldsymbol{\Phi}$ is a subsampling and scaling matrix, i.e., when $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$ consists of M rows of a diagonal matrix.[7] With this method we can still obtain an algorithm with reasonable computational complexity per iteration, assuming that $\hat{K}$ is relatively small (a $\hat{K} \times \hat{K}$ matrix must be inverted). We coin this algorithm as semifast. For small $\hat{K}$ the semifast algorithm is faster than the superfast algorithm of Sec. B.3 and it may therefore be beneficial to even use it in the complete data case.

The semifast method is based on the following decomposition of $\mathbf{C}^{-1}$, obtained using Woodbury's matrix identity:

$$\mathbf{C}^{-1} = \hat{\beta}^{-1}\mathbf{I} - \hat{\beta}^{-2}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\hat{\boldsymbol{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}}) \tag{B.43}$$

with $\hat{\boldsymbol{\Sigma}}$ given by (B.11). We can evaluate $\hat{\boldsymbol{\Sigma}}^{-1}$ by noting that

$$\begin{aligned}\left[\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\right]_{i,k} &= \left[\boldsymbol{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{z}})\right]_{i,k} \\ &= \sum_{m=1}^{M} |\boldsymbol{\Phi}_{m,I_{\mathcal{M}}(m)}|^2 \exp\left(j2\pi(I_{\mathcal{M}}(m)-1)(\hat{\theta}_k - \hat{\theta}_i)\right), \end{aligned} \tag{B.44}$$

which can be evaluated with a NUFFT in time $\mathcal{O}(N \log N + \hat{K}^2)$. Forming $\hat{\boldsymbol{\Sigma}}^{-1}$ is then easy and an inversion[8] in time $\mathcal{O}(\hat{K}^3)$ is needed to obtain $\hat{\boldsymbol{\Sigma}}$.

[7] Let $\mathcal{M} \subseteq \{1, \ldots, N\}$ denote the index set of the observed entries and $I_{\mathcal{M}} : \{1, \ldots, M\} \to \mathcal{M}$ be an indexing. Then $\boldsymbol{\Phi}_{m,I_{\mathcal{M}}(m)}$, $m = 1, \ldots, M$, are the only nonzero elements of $\boldsymbol{\Phi}$.

[8] As is customary in numerical linear algebra, we would recommend not to explicitly evaluate the inverse, but instead use the numerically stabler and faster approach of calculating the Cholesky decomposition $\hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{L}\mathbf{L}^{\mathrm{H}}$ (a unique Cholesky decomposition exists because $\hat{\boldsymbol{\Sigma}}^{-1}$ is Hermitian positive definite). We need to evaluate matrix-vector products involving $\hat{\boldsymbol{\Sigma}}$ which are easily evaluated from the decomposition by forward-backward substitution. We can also calculate $|\hat{\boldsymbol{\Sigma}}^{-1}|$ directly from the Cholesky decomposition.

The approach thus hinges on $\hat{K}$ being sufficiently small, such that the inverse (really, the Cholesky decomposition) can be calculated in reasonable time.

B.4.1 Evaluating $y^{\mathrm{H}}\mathbf{C}^{-1}y$, $\ln|\mathbf{C}|$, q, r and u

Notice that matrix-vector products involving $\mathbf{\Psi}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ and $\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ can be evaluated using a NUFFT. It then immediately follows that the values $y^{\mathrm{H}}\mathbf{C}^{-1}y$, q, r and u can be evaluated using (B.43) with complexity $\mathcal{O}(\hat{K}^2 + N\log N)$.

To evaluate the objective function (B.8) we need to calculate $\ln|\mathbf{C}|$. By invoking the matrix determinant lemma we get

$$\ln|\mathbf{C}| = M\ln\hat{\beta} + \sum_{\{k:\hat{z}_k=1\}} \ln\hat{\gamma}_k + \ln|\mathbf{\Sigma}^{-1}|, \tag{B.45}$$

which can be evaluated in time $\mathcal{O}(\hat{K})$ once the Cholesky decomposition of $\mathbf{\Sigma}^{-1}$ is known.

B.4.2 Evaluating s, t, v and x

As an example, we demonstrate how to evaluate t. We note that s, v and x can easily be obtained using the same approach. First, insert (B.43) into (B.19) to get

$$\begin{aligned} t = \hat{\beta}^{-1}\operatorname{diag}\left(\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\right) \\ - \hat{\beta}^{-2}\operatorname{diag}\left(\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\hat{\mathbf{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\right). \end{aligned}$$

Using the same methodology as for computing $\hat{\mathbf{\Sigma}}^{-1}$, the $\hat{K}\times\hat{K}$ matrices $\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ and $\mathbf{\Psi}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{D}\mathbf{\Phi}^{\mathrm{H}}\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})$ can be obtained in time $\mathcal{O}(N\log N + \hat{K}^2)$. Then, t is found by direct evaluation in time $\mathcal{O}(\hat{K}^3)$.

B.4.3 Evaluating $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta\in\mathcal{G}$

To calculate the frequency of the component processed in the activation stage we must evaluate $q_{\sim k}(\theta)$ and $s_{\sim k}(\theta)$ for all $\theta\in\mathcal{G}$, where $\mathcal{G}$ is a grid of L equispaced points. Using the fact that in the beginning of the activation step the kth component is deactivated and thus $\mathbf{C}_{\sim k} = \mathbf{C}$, we obtain the required quantities by inserting (B.43) into (B.16) and (B.18):

$$\begin{aligned} q^{\mathcal{G}} &= \hat{\beta}^{-1}\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\left(y - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\hat{\boldsymbol{\mu}}\right) \\ s^{\mathcal{G}} &= \hat{\beta}^{-1}\operatorname{diag}\left(\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})\right) \\ &\quad - \hat{\beta}^{-2}\operatorname{diag}\left(\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}^{\mathcal{G}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\hat{\mathbf{\Sigma}}\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})\right). \end{aligned}$$

It is clear that $\boldsymbol{q}^{\mathcal{G}}$ can easily be found using FFT techniques.

To obtain $\boldsymbol{s}^{\mathcal{G}}$ we first note that the first term is a vector with all entries equal to $\hat{\beta}^{-1}\sum_{m=1}^{M}|\boldsymbol{\Phi}_{m,I_{\mathcal{M}}(m)}|^2$. The second term is found by using a NUFFT (see (B.44)) to form $\mathbf{A}^{\mathrm{H}}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{z}}})\mathbf{A}(\boldsymbol{\theta}^{\mathcal{G}})$. Then by using the Cholesky decomposition of $\hat{\boldsymbol{\Sigma}}^{-1}$ the second term can be calculated in time $\mathcal{O}(L\hat{K}^2)$.

B.4.4 Algorithm Complexity

The above computation is dominated by either the calculation of $\boldsymbol{s}^{\mathcal{G}}$ or the length-L FFT involved in calculating $\boldsymbol{q}^{\mathcal{G}}$. Again with $L = 8N$ we have overall complexity per iteration $\mathcal{O}(N\hat{K}^2 + N\log N)$.

B.5 Multiple Measurement Vectors

The algorithm presented in Sec. B.2 assumes a single measurement vector (SMV). We now discuss an extension to the case of multiple measurement vectors (MMV) [34]. This case is of particular importance in array processing where the number of observation points M is determined by the number of antennas in the array.[9] Typically M is small, which thus limits estimation accuracy. On the other hand it is often easy to obtain multiple observation vectors across which the entries in $\tilde{\boldsymbol{\theta}}$ (the true directions of arrivals) are practically unchanged. The MMV signal model reads

$$\boldsymbol{y}^{(g)} = \mathbf{A}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}^{(g)} + \boldsymbol{w}^{(g)}, \tag{B.46}$$

where $g = 1,\ldots,G$ indexes the observation vectors.

To extend our SMV algorithm to the MMV case we again impose an estimation model of the form (B.2) that contains $K_{\max}$ components which can be (de)activated based on variables z_k, $k = 1,\ldots,K_{\max}$. The likelihood for each of the G observation vectors then reads

$$p(\boldsymbol{y}^{(g)}|\boldsymbol{\alpha}^{(g)},\boldsymbol{z},\boldsymbol{\theta};\beta) = \mathrm{CN}(\boldsymbol{y}^{(g)};\mathbf{A}(\boldsymbol{\theta}_{\boldsymbol{z}})\boldsymbol{\alpha}_{\boldsymbol{z}}^{(g)},\beta\mathbf{I}). \tag{B.47}$$

We impose the same prior as used in the SMV case (B.5) on each $\boldsymbol{\alpha}^{(g)}$:

$$p(\boldsymbol{\alpha}^{(g)};\boldsymbol{\gamma}) = \prod_{k=1}^{K_{\max}}\mathrm{CN}(\alpha_k^{(g)};0,\gamma_k). \tag{B.48}$$

The vectors $\boldsymbol{z}$ and $\boldsymbol{\theta}$ are assigned the same priors as in the SMV case, i.e., as given by (B.4) and (B.6). Similarly to the SMV case, the MMV model has parameters γ, β and ζ.

[9] It is worth noting that in array processing the complete data case corresponds to the very common situation of using a uniform linear array.

The objective to be minimized is the marginal likelihood, which for the MMV model reads

$$\begin{aligned}\mathcal{L}_{\text{MMV}} &\triangleq -\ln \prod_{g=1}^{G} p(\boldsymbol{y}^{(g)}|\boldsymbol{z},\boldsymbol{\theta};\beta,\gamma)p(\boldsymbol{z};\zeta)p(\boldsymbol{\theta}) + \text{const.},\\ &= \sum_{g=1}^{G}\left[\ln|\mathbf{C}| + \left(\boldsymbol{y}^{(g)}\right)^{\text{H}}\mathbf{C}^{-1}\boldsymbol{y}^{(g)}\right]\\ &\qquad - \sum_{k=1}^{K_{\max}}\left(z_k \ln\zeta + (1-z_k)\ln(1-\zeta)\right) + \text{const.},\end{aligned}$$

where $p(\boldsymbol{y}^{(g)}|\boldsymbol{z},\boldsymbol{\theta};\beta,\gamma) = \text{CN}(\boldsymbol{y}^{(g)};\mathbf{0},\mathbf{C})$ with $\mathbf{C}$ as in (B.7). The posterior probabilities of the coefficient vectors $\boldsymbol{\alpha}^{(g)}$, $g = 1,\ldots,G$, are given by (B.9) with $\boldsymbol{y}$ and $\boldsymbol{\alpha}$ replaced by $\boldsymbol{y}^{(g)}$ and $\boldsymbol{\alpha}^{(g)}$.

The procedure to estimate the variables $\boldsymbol{\theta}$, $\boldsymbol{z}$, γ, β and ζ follows straightforwardly from the method used in the SMV case. Here we provide a brief discussion of the derivation of the update equations; refer to Sec. B.2 for details.

To estimate $\boldsymbol{\theta}$ and γ the first- and second-order derivatives of $\mathcal{L}_{\text{MMV}}$ are needed. Denote the derivative (B.12) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$ as $\frac{\partial\mathcal{L}^{(g)}}{\partial[\boldsymbol{\theta}_{\hat{z}}]_k}$. Then we have

$$\frac{\partial\mathcal{L}_{\text{MMV}}}{\partial[\boldsymbol{\theta}_{\hat{z}}]_k} = \sum_{g=1}^{G}\frac{\partial\mathcal{L}^{(g)}}{\partial[\boldsymbol{\theta}_{\hat{z}}]_k}.$$

A similar result follows for the second-order derivative and the derivatives with respect to $\gamma_{\hat{z}}$.

The estimate of ζ is unchanged from the SMV case (B.23).

To estimate the noise variance β, we write an upper bound of the same form as (B.25) and find its minimizer to be

$$\begin{aligned}\hat{\beta} = \max\Big(\varepsilon_\beta, M^{-1}\,\text{tr}\left(\hat{\boldsymbol{\Sigma}}\mathbf{A}^{\text{H}}(\hat{\boldsymbol{\theta}}_{\hat{z}})\mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\right)\\ + (GM)^{-1}\sum_{g=1}^{G}||\boldsymbol{y}^{(g)} - \mathbf{A}(\hat{\boldsymbol{\theta}}_{\hat{z}})\hat{\boldsymbol{\mu}}^{(g)}||^2\Big),\end{aligned}$$

where $\hat{\boldsymbol{\mu}}^{(g)}$ is given by (B.10) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$.

To write the activation and deactivation criteria for the MMV model we rewrite the objective in terms of the parameters of a single component, anal-

ogously to (B.27):

$$\mathcal{L}_{\mathrm{MMV}}(z_k, \theta_k, \gamma_k) = z_k \Bigg(G \ln\big(1 + \gamma_k s_{\sim k}(\theta_k)\big) - \sum_{g=1}^{G} \frac{|q_{\sim k}^{(g)}(\theta_k)|^2}{\gamma_k^{-1} + s_{\sim k}(\theta_k)} + \ln\left(\frac{1-\hat{\zeta}}{\hat{\zeta}}\right) \Bigg) + \text{const.}, \quad \text{(B.49)}$$

where $q_{\sim k}^{(g)}(\theta_k)$ is given by (B.28) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^{(g)}$. We omit the details of the activation and deactivation stages, as they follow straightforwardly from (B.49) and the description in Secs. B.2.3 and B.2.3.

The insightful reader may have noticed that the calculations required for MMV are very similar to those required for SVM. In particular, the matrix $\hat{\mathbf{C}}$ is unchanged and the methods for calculating matrix-vector products involving $\hat{\mathbf{C}}^{-1}$ presented in Secs. B.3 and B.4 can be utilized. All expressions involving $\boldsymbol{y}$ (i.e., $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{u}$, $\boldsymbol{q}^{\mathcal{G}}$ and $\boldsymbol{y}\hat{\mathbf{C}}^{-1}\boldsymbol{y}$) must be calculated for each observation vector $\boldsymbol{y}^{(g)}$. This means that in the case of complete observations, the generalized Schur algorithm can be used so that the MMV algorithm has per-iteration complexity $\mathcal{O}(N \log^2 N + GN \log N)$. With incomplete observations the semifast method can be used with per-iteration complexity $\mathcal{O}(N\hat{K}^2 + GN \log N)$.

B.6 Experiments

B.6.1 Setup, Algorithms & Metrics

In our experiments we use the signal model (B.1). In the following the wrap-around distance on $[0, 1)$ is used for all differences of frequencies (see Footnote 5). Unless otherwise noted, the true frequencies are drawn randomly, such that the minimum separation between any two frequencies is $2/N$. Specifically, the frequencies are generated sequentially for $k = 1, \ldots, K$ with the kth frequency, $\tilde{\theta}_k$, drawn from a uniform distribution on the set $\{\theta \in [0, 1) : d(\theta, \tilde{\theta}_l) > 2/N \text{ for all } l < k\}$.

The true coefficients in $\tilde{\boldsymbol{\alpha}}$ are generated i.i.d. random, with each entry drawn as follows. First a circularly-symmetric complex normal random variable a_k with standard deviation 0.8 is drawn. The coefficient is then found as $\tilde{\alpha}_k = a_k + 0.2\, e^{j \arg(a_k)}$. The resulting random variable has the property $|\tilde{\alpha}_k| \geq 0.2$, i.e., all components have significant magnitude. We use this specification to ensure that all components can be distinguished from noise. After generating the set of K frequencies and coefficients, the noise variance β is selected such that the desired signal-to-noise ratio (SNR) is obtained, with $\mathrm{SNR} \triangleq ||\boldsymbol{\Phi}\boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}||^2/(M\beta)$.

We compare the superfast LSE algorithm[10] with the following reference algorithms: variational Bayesian line spectral estimation (VALSE) [16]; atomic soft thresholding[11] (AST) [12]; gridless SPICE[12] (GLS) [35]; ESPRIT [6, 8]; and a gridded solution obtained with the least absolute shrinkage and selection operator (LASSO) solved using SpaRSA[13] [36].

The solution to the primal problem of AST [12] directly provides an estimate of the signal vector $\boldsymbol{h} = \boldsymbol{\Psi}(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\alpha}}$. This solution is known to be biased towards the all-zero solution (as is also the case with the classical LASSO solution). A so-called *debiased* solution can be obtained by recovering the frequencies from the AST dual and estimating the coefficients $\tilde{\boldsymbol{\alpha}}$ via least-squares. As in [12], we report here the debiased solution. If the frequencies are separated by at least $2/N$, the AST algorithm is known to exactly recover the frequencies in the noise-free case [11–13]. In the noisy case no such recovery guarantee exists, but a bound on the estimation error of the signal vector $\boldsymbol{h}$ is known [12, 13]. Unfortunately this error bound does not apply to the debiased solution we report herein.

We use the variant of GLS [35] which uses SORTE [9] for model order estimation and MUSIC [7] for frequency estimation.

ESPRIT requires an estimate of the signal covariance matrix and of the model order. The former is obtained as the averaged sample covariance matrix computed from the signal vector split into $N/3$ signal vectors of length $2N/N$ using forward-backward smoothing. The model order is estimated with SORTE [9].

The LASSO solution is obtained using a grid of size $8N$. We have observed that no improvement in performance is achieved with a finer grid. The regularization parameter of LASSO is selected as proposed in [12] with knowledge of the true noise variance. We use the debiased solution returned by the SpaRSA solver.

In the evaluation of the signal reconstruction we have also included an oracle estimator (denoted Oracle) which obtains a least squares solution for $\tilde{\boldsymbol{\alpha}}$ with known $\tilde{\boldsymbol{\theta}}$.

Three performance metrics are used: normalized mean-squared error (NMSE) of the reconstructed signal, block success rate (BSR) and component success rate (CSR). The NMSE is defined as

$$\text{NMSE} \triangleq \frac{||\boldsymbol{\Psi}(\hat{\boldsymbol{\tau}})\hat{\boldsymbol{\alpha}} - \boldsymbol{\Psi}(\tilde{\boldsymbol{\tau}})\tilde{\boldsymbol{\alpha}}||^2}{||\boldsymbol{\Psi}(\tilde{\boldsymbol{\tau}})\tilde{\boldsymbol{\alpha}}||^2}.$$

[10] We have published our code at github.com/thomaslundgaard/superfast-lse. It is based on our own implementation of the generalized Schur algorithm and the NUFFT available at cims.nyu.edu/cmcl/nufft/nufft.html.

[11] The code is available online at github.com/badrinarayan/astlinespec.

[12] The code has kindly been provided by the authors.

[13] The code is available online at lx.it.pt/∼mtf/SpaRSA.

The BSR is the proportion of Monte Carlo trials in which the frequency vector $\tilde{\boldsymbol{\theta}}$ is successfully recovered. Successful recovery is understood as correct estimation of the model order K and that $||d(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})||_\infty < 0.5/N$. The association of the entries in $\hat{\boldsymbol{\theta}}$ to those in $\tilde{\boldsymbol{\theta}}$ is obtained by using the Hungarian method [37] (also known as Munkres assigment algorithm) minimizing $||d(\tilde{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}})||_2^2$.

The BSR can be misleading, since a trial is considered to be unsuccessful if just a single component is misestimated; for example if a component is represented in the estimate by two components with very close frequencies. We therefore introduce the CSR, defined as follows:

$$\text{CSR} \triangleq \frac{\sum_{k=1}^{\hat{K}} S(\hat{\theta}_k, \tilde{\boldsymbol{\theta}}) + \sum_{k=1}^{K} S(\tilde{\theta}_k, \hat{\boldsymbol{\theta}})}{\hat{K} + K}$$

with the success function $S(x, \boldsymbol{a}) \triangleq \mathbb{1}[\min_k d(x, a_k) < 0.5/N]$, where $\mathbb{1}[\cdot]$ denotes the indicator function. The reported CSR is averaged over a number of Monte Carlo trials. The CSR takes values in $[0, 1]$. A CSR of 1 is achieved if, and only if, all estimated components are in the vicinity of one or more true components and all true components are in the vicinity of one or more estimated components.

B.6.2 Choosing the Activation Threshold

To determine a sensible value for the activation threshold τ in (B.34), the following experiment is conducted. We consider the complete data case with $N = M = 128$ and the number of components is fixed at $K = 35$, as there is a larger tendency to activate artefact components for relatively large K/N. The algorithm is provided with the knowledge of $K_{\max} = 35$ and the activation probability is fixed at $\hat{\zeta} = 35/128$. The algorithm is run with the activation criterion (B.33). In this way, the algorithm in most cases successfully estimates the frequencies without any artefacts. After the algorithm has terminated, we test if $\tilde{\boldsymbol{\theta}}$ was successfully recovered (as defined above). If so, $K_{\max}$ is increased and the procedure for activating a component in Sec. B.2.3 is run and the difference between the left-hand and right-hand sides of (B.33) is saved. We refer to this difference as δ and the criterion (B.33) can be expressed as $\delta > 0$. In Fig. B.1 we show histograms of the value δ obtained from $5,000$ successful recoveries at three different SNR values At each SNR, the experiment is repeated until the required number of successful recoveries are obtained; trials without successful recovery are discarded. Cases where $\delta > 0$ thus correspond to cases where an artefact would be activated using criterion (B.33).

The heuristic criterion (B.34) corresponds to $\delta > \tau$. From Fig. B.1 it is clearly seen that threshold $\tau = 5$ is a sensible value, which precludes almost

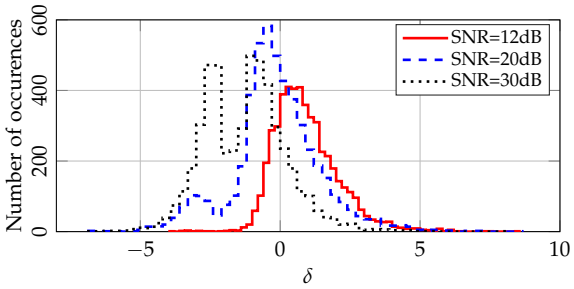

Fig. B.1: Histograms of δ values for different signal-to-noise ratios. Cases where $\delta > 0$ correspond to cases where an artefact component is activated if the criterion (B.33) is used.

all artefact components from being activated. It is seen that this threshold works well for a large range of SNR values. We have not investigated whether $\tau = 5$ is so large that desired components are precluded from activation. The results in the following investigations are all obtained with $\tau = 5$ and the good performance of our algorithm across this wide selection of scenarios indicates that the selected τ is not too large.

B.6.3 Estimation with Complete Data

In Fig. B.2 we show performance results versus SNR. We first notice that Superfast LSE is on par with or outperforms all other algorithms in the three metrics shown here for all SNR values. In the low SNR region no algorithm can reliably recover the correct model order and the frequencies. In the plots of the CSR and NMSE, we see that Superfast LSE, VALSE and AST generally achieve the best approximation of the true frequencies. There is a small performance gap in terms of NMSE between Oracle and all other evaluated algorithms due to the uncertainty in frequency estimation (Oracle knows the true frequencies).

ESPRIT and GLS are observed to have the weakest performance at low SNR, especially in terms of CSR and NMSE. Both algorithms use SORTE to estimate the model order from the eigenvalues of the signal covariance. At low SNR it is hard to distinguish the signal eigenvalues from the noise eigenvalues, leading to the observed deterioration in performance.

At medium to high SNR, BSR of AST is about 0.75. The algorithm tends to slightly overestimate the model order (not shown here). We hypothesise that such systematic overestimation of the model order can be avoided by adjusting the regularization parameter used in AST. Doing so would, however, mean that AST would perform worse in other scenarios. This is exactly the weakness of methods involving regularization parameters.

Finally note that LASSO is never able to successfully estimate the model order, due to the use of a grid. In particular each true frequency component

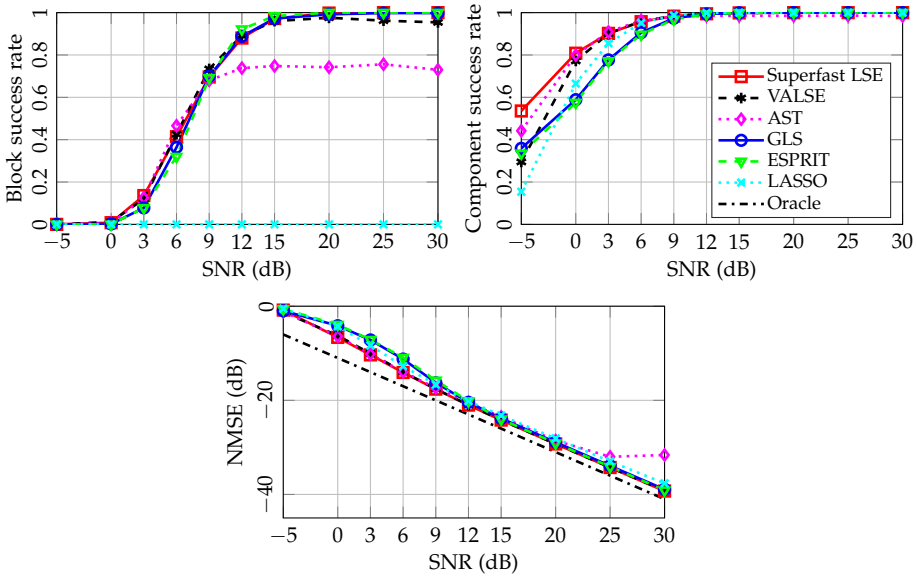

Fig. B.2: Simulation results for varying SNR with complete data ($\mathbf{\Phi} = \mathbf{I}$). The signal length is $N = M = 128$ and the number of components is $K = 10$. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

is estimated by a few non-zero entries at neighbouring gridpoints. It is visible in the CSR that LASSO indeed estimates frequencies which lie in the vicinity of the true frequencies. In some applications, e.g. channel estimation in wireless communications, it is the reconstructed signal and not the frequencies themself which are of interest. In this case LASSO may be preferable because of its simplicity. Due to the grid approximation, LASSO performs a little worse than the best gridless algorithms in terms of NMSE.

B.6.4 Super Resolution

The ability to separate components beyond the Rayleigh limit of $1/N$ is known as super resolution. The results in Fig. B.3 illustrate the super resolution ability of the algorithms. In this experiment we generate 5 pairs of frequencies with varying distance between the two frequencies in each pair. The pairs are well separated (at least $2/N$ separation between frequencies which are not in the same pair).

The NMSE performance of Superfast LSE, VALSE and LASSO is only slightly worse at low separation when compared to the performance at large separation. It is evident that the model order and the frequencies cannot be recovered in every case (BSR below 1) when the separation is less than $1/N$.

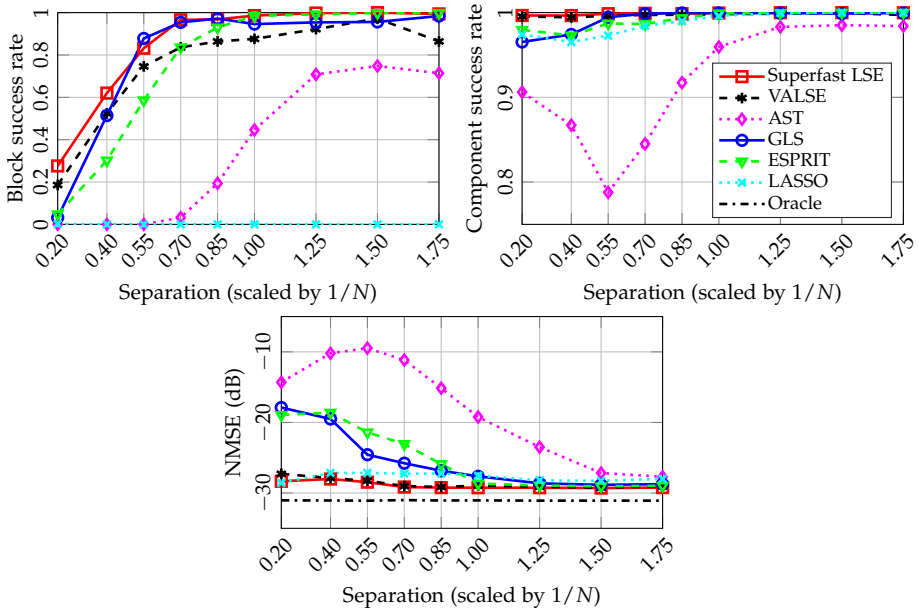

Fig. B.3: Simulation results for closely23 located components with complete data ($\mathbf{\Phi} = \mathbf{I}$). The frequencies are generated as 5 pairs (i.e. $K = 10$) such that each pair has varying (deterministic) intra-pair separation, while the location of the pairs are generated randomly with non-paired frequencies at least $2/N$ apart (i.e., the location of the pairs are generated using a procedure similar to the one described in Sec. B.6.1). The signal length is $N = M = 128$ and the SNR is 20 dB. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

Since the CSR is close to 1 and the NMSE is close to that of Oracle, these three algorithms handle closely located components well, in the sense that a good approximation of the frequencies is obtained.

AST, GLS and ESPRIT give a CSR below 1 and a rather large NMSE when the separation is small. This is despite the fact that GLS and ESPRIT do not show a significantly worse BSR compared to Superfast LSE. We have observed that this is because these algorithms significantly underestimate the model order in some cases, resulting in a large contribution to NMSE.

ESPRIT shows worse super resolution ability than Superfast LSE, VALSE and GLS (lower BSR for separation below $0.7/N$). This is because a covariance matrix of size $2N/3$ is formed, thus reducing the effective signal length.

B.6.5 Estimation with Incomplete Data

Fig. B.4 reports the performance in the incomplete data case. The measurement matrix $\mathbf{\Phi}$ is generated by randomly selecting M rows of the $N \times N$

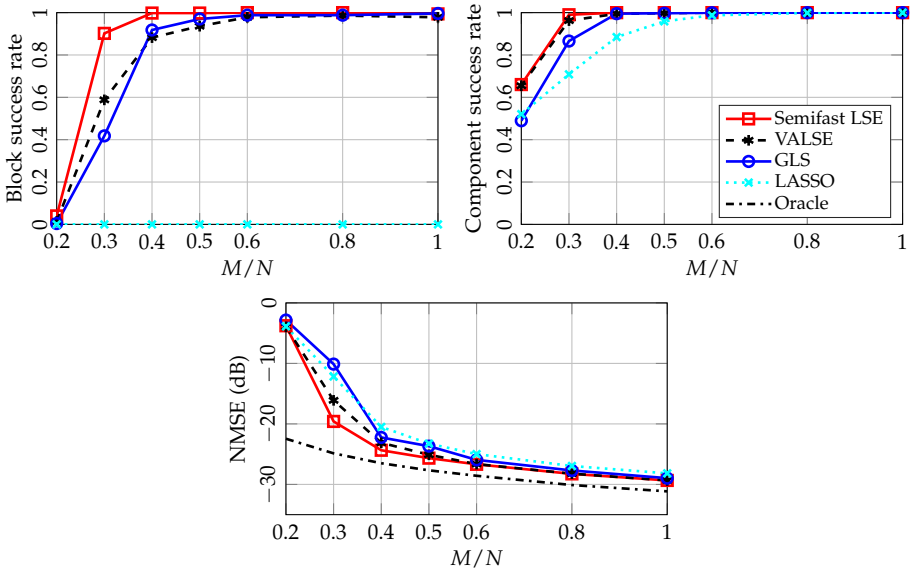

Fig. B.4: Simulation results with incomplete data, i.e., $\mathbf{\Phi}$ contains M rows of $\mathbf{I}$ selected at random. The signal length is $N = 128$, the SNR is 20 dB and the number of components is $K = 10$. Results are averaged over 500 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

identity matrix. The set of observation indices is chosen to include the first and last indices, while the remaining $M - 2$ indices are obtained by uniform random sampling without replacement. Only a subset of the algorithms are applicable in this case. Our proposed algorithm is implemented using the techniques described in Sec. B.4. We refer to it as Semifast LSE.

Semifast LSE and VALSE largely show the same performance, while GLS has a slightly higher NMSE for $M/N \leq 0.5$. The higher NMSE is caused by a few outliers (less than 1% of the Monte Carlo trials) where GLS significantly underestimates the model order. LASSO is again observed to have reasonable NMSE and CSR while being unable to correctly estimate the set of frequencies (i.e., BSR$= 0$).

B.6.6 Phase Transitions

Inspired by the concept of phase transitions in compressed sensing, we perform an experiment which shows a similar phenomenon for LSE. In particular we demonstrate that for each algorithm there is a region in the space of system parameters where it can almost perfectly recover the frequencies and a region where it cannot, with a fairly sharp transition between the two. The results, in terms of BSR, are seen in Fig. B.5.

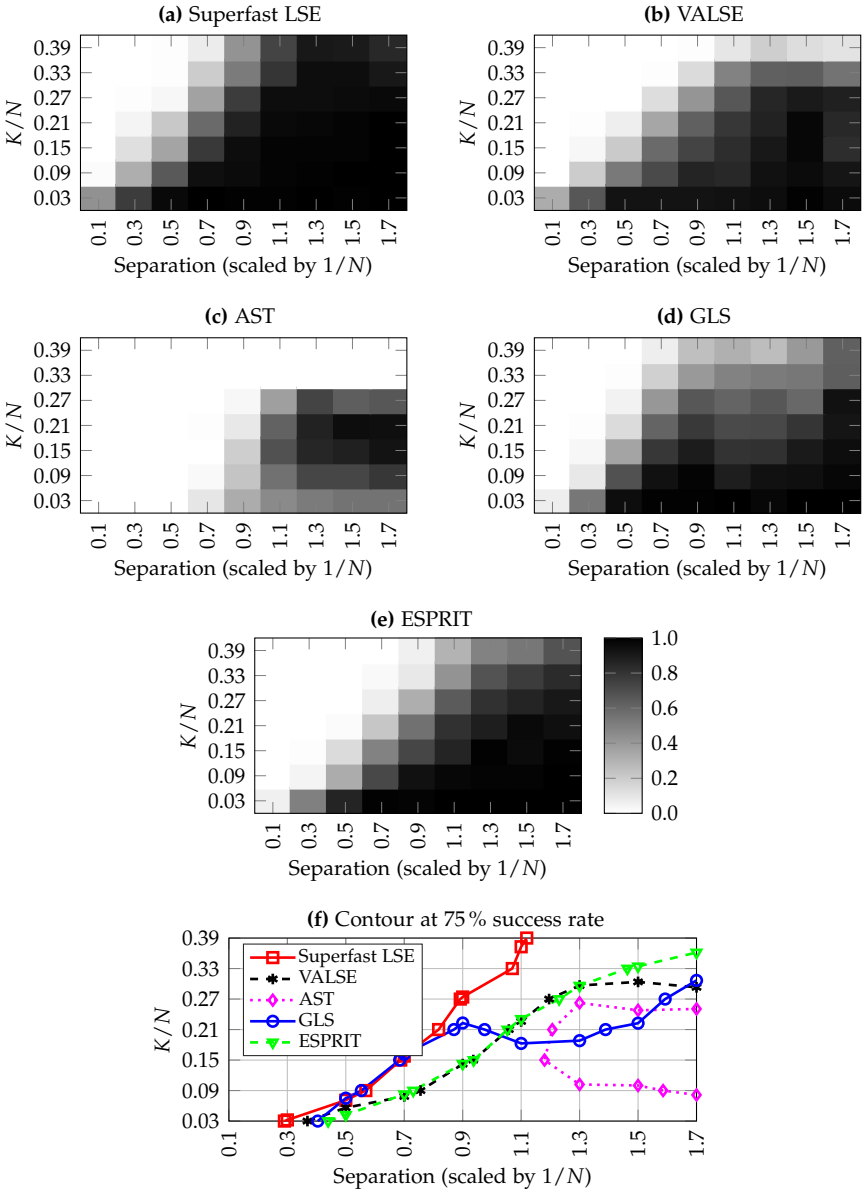

Fig. B.5: Simulation results showing phase transitions with complete data ($\mathbf{\Phi} = \mathbf{I}$). The plots show block success rate. The set of frequencies are generated as closely located pairs, following the methodology described in the caption of Fig. B.3. The number of pairs are selected such that the specified ratio K/N is achieved as closely as possible. The signal length is $N = M = 128$ and the SNR is 20 dB. Results are averaged over 120 Monte Carlo trials.

We first note that AST has rather poor performance, which is consistent with the observation in Fig. B.2 that its BSR is significantly below 1. Turning our attention to VALSE, GLS and ESPRIT, we see that these algorithms generally do not deal well with a large number of components, in the sense that the BSR is significantly below 1 for $K/N \geq 0.15$. It is seen in Fig. B.5f that Superfast LSE has the largest region with high probability of successful recovery (BSR ≥ 0.75).

B.6.7 Computation Times

In Fig. B.6 and B.7 we show algorithm runtimes for varying problem sizes. Our proposed method uses the superfast and semifast implementations in Secs. B.3 and B.4, respectively. The results are obtained using MATLAB on a 2011 MacBook Pro. To avoid differences in results originating from the amount of parallelism achieved in each implementation, MATLAB is restricted to only use a single computational thread. The part of the code where each algorithm spends significant time is implemented as native code via MATLAB's codegen feature.

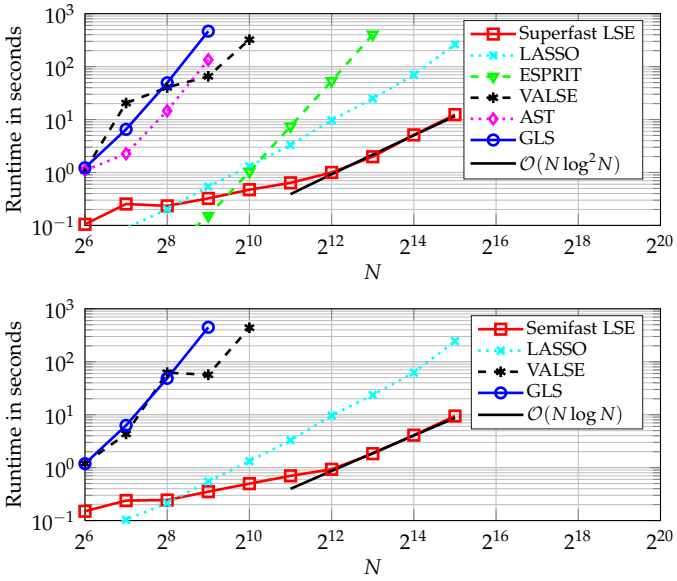

Fig. B.6: Runtimes in seconds versus problem size N. We show for both complete (top) and incomplete (bottom) data case. The number of components is $K = 15$ and the SNR is 20 dB. Values are averaged over 20 Monte Carlo trials. In the incomplete data case we set $M = 0.75N$. We also plot (solid black) the asymptotic per-iteration complexity of Superfast and Semifast LSE.

For varying N (Fig. B.6), we first observe that for small to moderate problem sizes ($N \leq 2^{10}$) the difference between LASSO and our proposed

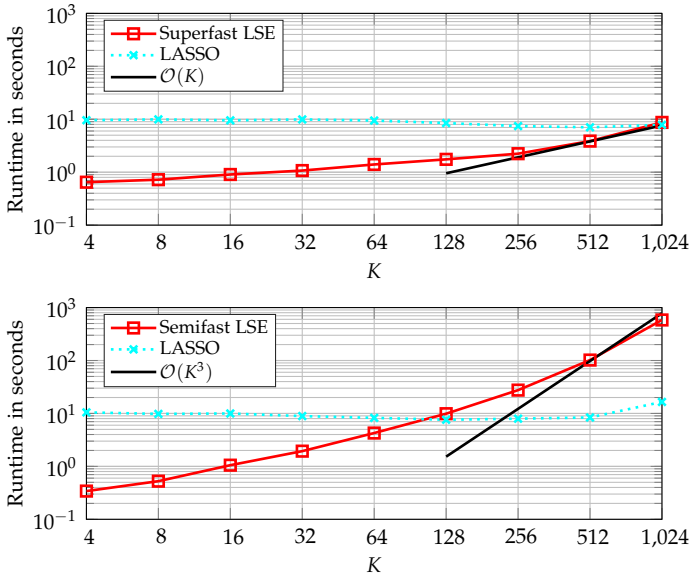

Fig. B.7: Runtimes in seconds versus number of components K. We show for both complete (top) and incomplete (bottom) data case. The problem size is $N = 4096$ and the SNR is 20 dB. Values are averaged over 20 Monte Carlo trials. We only show results for algorithms which has runtime lower than 10 s at $K = 4$. In the incomplete data case we set $M = 0.75N = 3072$. We also plot (solid black) the asymptotic per-iteration complexity of Superfast and Semifast LSE.

algorithms is small (less than 1 second). This difference is mainly due to implementation details. In the large-N region, Superfast and Semifast LSE are approximately an order of magnitude faster than LASSO. We observe that the asymptotic per-iteration complexity of Superfast and Semifast LSE describes the scaling of the total runtime well for $N \geq 2^{12}$, because the number of iterations (not shown) stays practically constant. The state-of-the-art LSE methods VALSE, AST and GLS all have $\mathcal{O}(N^3)$ time complexity or worse, which results in very large runtimes even when the problem size is moderate (e.g. > 100 s for AST and GLS at $N = 512$). For large problem sizes, the $\mathcal{O}(N^3)$ time complexity of ESPRIT is evident and Superfast/Semifast LSE and LASSO significantly outperform ESPRIT.

In Fig. B.7 we show results illustrating how the computation time scales with K. In this analysis we assume $\hat{K} = \mathcal{O}(K)$. First we note that the runtime of LASSO is practically constant with K. In the complete data case, the per-iteration complexity of Superfast LSE scales linearly with K. In practice we see a slower scaling with K, which means that the values of K we use here are not large enough to reach the asymptotic region. Simulations with large K cannot be run, because we need approximately $K < N/4$ for $\hat{K} = \mathcal{O}(K)$ to

hold (cf. Fig. B.5).

In the incomplete data case, we see that the runtime of Semifast LSE increases quickly with K, such that for $K > 128$ LASSO is faster than Semifast LSE. We do, however, see that the asymptotic complexity of $\mathcal{O}(K^3)$ is not reached in our experiment, because the runtime is dominated by the calculation of $\boldsymbol{s}^{\mathcal{G}}$, which has complexity $\mathcal{O}(LK^2)$.

B.7 Conclusions

We have presented a low-complexity algorithm for line spectral estimation. Computational methods for both the complete and incomplete data cases have been presented, along with an extension to the case of multiple measurement vectors.

The proposed algorithm falls in the category of Bayesian methods for line spectral estimation. Bayesian methods are widely accepted due to their high estimation accuracy, but a drawback of this class of methods has historically been their large computational complexity. In that respect, this work makes an important contribution in making Bayesian methods more viable in practice.

At the core of the computational method for the complete data case lies the application of the Gohberg-Semencul formula to the Toeplitz signal covariance matrix. Many methods for line spectral estimation have Toeplitz covariance matrices at their core and we conjecture that the computational complexity of some of them can be drastically reduced by applying the techniques we have demonstrated in this paper.

Our numerical experiments show that our Superfast LSE algorithm has very high estimation accuracy. For example, in Fig. B.5 we see that Superfast LSE attains high frequency recovery rates for a much larger set of scenarios than any of the reference algorithms. At the same time our algorithm has so low computation time that it makes highly-accurate LSE feasible for problems with size much larger than methods currently available in the literature can practically deal with.

A Convergence Analysis

We now discuss the convergence of our proposed block-coordinate descent algorithm. To do so we introduce an iteration index i on all estimated variables. Our algorithm then produces sequences of blocks of estimates denoted as $\{\hat{\boldsymbol{z}}^i\}$, $\{\hat{\boldsymbol{\zeta}}^i\}$, $\{\hat{\boldsymbol{\beta}}^i\}$ and $\{(\hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i)\}$. We denote the value of the objective function at the end of the ith iteration as $\mathcal{L}^i \triangleq \mathcal{L}(\hat{\boldsymbol{z}}^i, \hat{\boldsymbol{\zeta}}^i, \hat{\boldsymbol{\beta}}^i, \hat{\boldsymbol{\theta}}^i, \hat{\boldsymbol{\gamma}}^i)$. We first note that all updates of the estimates are guaranteed not to increase the objective, so $\{\mathcal{L}^i\}$ is a non-increasing sequence. Since $\boldsymbol{\beta} \geq \varepsilon_{\boldsymbol{\beta}}$ it is also bounded below

and thus it converges. Therefore, our proposed algorithm terminates in finite time.

Unfortunately, convergence of $\{\mathcal{L}^i\}$ does not imply convergence of the sequences of estimates. Even with exact minimization in each block of variables, block-coordinate descent on non-convex functions can get stuck in an infinite cycle [38]. This is further complicated by the fact that our algorithm only approximately solves the minimization in some of the blocks.

Proposition 5 in [39] shows that with exact minimization in each block, block-coordinate descent converges if the objective function is strictly quasiconvex in all but 2 blocks. The objective function $\mathcal{L}$ is strictly quasiconvex in ζ and β. There is thus hope that we can prove convergence of our scheme which, in lieu of computing the exact minimizer, merely has a descent property in each block. Our approach to show convergence shares the same overall idea as that in [39], while many details differ.

To discuss the convergence properties, we first derive a number of lemmas. Theorem B.1 is proved at the end of the appendix. For notational simplicity we take the convention that for each i the block-coordinate descent algorithm cycles through the block updates in the following order: $\hat{z}^i, \hat{\zeta}^i, \hat{\beta}^i$ and finally $(\hat{\boldsymbol{\theta}}^i, \hat{\gamma}^i)$, such that for example $\hat{\beta}^i$ is found based on ($\hat{z}^i$, $\hat{\zeta}^i$, $\hat{\beta}^{i-1}$, $\hat{\boldsymbol{\theta}}^{i-1}$, $\hat{\gamma}^{i-1}$). This is strictly speaking not how we have defined our algorithm, but that does not affect the correctness of our analysis.

Lemma B.1. *The sequence of estimates has at least one convergent subsequence, i.e., at least one limit point.*

Proof: All variables but γ and β are defined to be in a closed and bounded set. Since $\lim_{\beta\to\infty} \mathcal{L} = \infty$ we can restrict β to a closed and bounded set determined by the (finite) initial value of the objective function. A similar argument holds for each γ_k. The lemma then follows from the Bolzano-Weierstrass theorem. ■

Lemma B.2. *The sequence $\{\hat{z}^i\}$ converges.*

Proof: Each activation of a component gives a decrease in $\mathcal{L}$ of at least $\varepsilon_{\mathcal{L}}$. Since $\{\mathcal{L}^i\}$ is lower bounded, there can only be finitely many activations. Since there cannot be more deactivations than activations, also the number of deactivations is finite. There is thus a finite number of changes to $\hat{z}$ and $\{\hat{z}^i\}$ converges. We denote the limit point as $\bar{z}$. ■

Lemma B.3. *The sequence $\{\hat{\zeta}^i\}$ converges. Further, the limit point $\bar{\zeta}$ is the unique global minimizer of $\zeta \mapsto \mathcal{L}(\bar{z}, \zeta, \beta, \boldsymbol{\theta}, \gamma)$ for any β, $\boldsymbol{\theta}$ and γ.*

Proof: The first statement follows from Lemma B.2 since $\hat{\zeta}^i$ (B.23) is only a function of $\hat{z}^i$. The second statement results from the fact that $\hat{\zeta}^i$ is defined as the global minimizer of $\zeta \mapsto \mathcal{L}(\hat{z}^i, \zeta, \hat{\beta}^{i-1}, \hat{\boldsymbol{\theta}}^{i-1}, \hat{\gamma}^{i-1})$ and this global minimizer does not depend on $\hat{\beta}^{i-1}$, $\hat{\boldsymbol{\theta}}^{i-1}$ and $\hat{\gamma}^{i-1}$. ■

Lemma B.4. *The sequence $\{\hat{\beta}^i\}$ converges to the limit point $\bar{\beta}$. Further, for every limit point $(\bar{z}, \bar{\zeta}, \bar{\theta}, \bar{\gamma})$ of the remaining variables, the limit point $\bar{\beta}$ is a local minimum at the boundary ε_β or a stationary point of $\beta \mapsto \mathcal{L}(\bar{z}, \bar{\zeta}, \beta, \bar{\theta}, \bar{\gamma})$.*

Proof: To perform this proof we expand our previous notation and denote the upper bound (B.25) as $Q(\beta; \hat{z}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1})$. Then,

$$\begin{aligned}\mathcal{L}(\hat{z}^i, \hat{\zeta}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) &= Q(\hat{\beta}^{i-1}; \hat{z}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) \\ &\geq Q(\hat{\beta}^i; \hat{z}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) \geq \mathcal{L}(\hat{z}^i, \hat{\zeta}^i, \hat{\beta}^i, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}).\end{aligned}$$

Recalling that $\{\mathcal{L}^i\}$ converges, we have

$$\lim_{i\to\infty} \left| \mathcal{L}(\hat{z}^i, \hat{\zeta}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) - \mathcal{L}(\hat{z}^i, \hat{\zeta}^i, \hat{\beta}^i, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) \right| = 0,$$

and thus

$$\lim_{i\to\infty} \left| Q(\hat{\beta}^{i-1}; \hat{z}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) - Q(\hat{\beta}^i; \hat{z}^i, \hat{\beta}^{i-1}, \hat{\theta}^{i-1}, \hat{\gamma}^{i-1}) \right| = 0. \tag{B.50}$$

Reasoning by contradiction, assume that the sequence of estimates $\{\hat{\beta}^i\}$ has two limit points $\bar{\beta}_1$ and $\bar{\beta}_2$, such that $\bar{\beta}_1 \neq \bar{\beta}_2$. Let $(\bar{\theta}, \bar{\gamma})$ be any limit point of $\{\hat{\theta}^i, \hat{\gamma}^i\}$ (such a limit point exists due to Lemma B.1). Then by (B.50) we must have

$$Q(\bar{\beta}_1; \bar{z}, \bar{\beta}_1, \bar{\theta}, \bar{\gamma}) = Q(\bar{\beta}_2; \bar{z}, \bar{\beta}_1, \bar{\theta}, \bar{\gamma}). \tag{B.51}$$

Recalling the definition of $\hat{\beta}^i$, we have that $\bar{\beta}_2$ uniquely minimizes Q. Then, since we assumed $\bar{\beta}_1 \neq \bar{\beta}_2$, we get

$$Q(\bar{\beta}_1; \bar{z}, \bar{\beta}_1, \bar{\theta}, \bar{\gamma}) > Q(\bar{\beta}_2; \bar{z}, \bar{\beta}_1, \bar{\theta}, \bar{\gamma}),$$

which contradicts (B.51). So $\{\hat{\beta}^i\}$ has only a single limit point which we denote as $\bar{\beta}$.

To prove the second statement, use (B.24) to show that

$$\frac{\partial}{\partial\beta} Q(\beta; \bar{z}, \bar{\beta}, \bar{\theta}, \bar{\gamma})\Big|_{\beta=\bar{\beta}} = \frac{\partial}{\partial\beta} \mathcal{L}(\bar{z}, \bar{\zeta}, \beta, \bar{\theta}, \bar{\gamma})\Big|_{\beta=\bar{\beta}}.$$

If $\bar{\beta} = \varepsilon_\beta$, we have that the derivatives of Q and thus of $\mathcal{L}$ are positive. It follows that $\bar{\beta}$ is a local or global minimum. If $\bar{\beta} \neq \varepsilon_\beta$ it is, by definition, a stationary point of Q. It is therefore also a stationary point of $\beta \mapsto \mathcal{L}(\bar{z}, \bar{\zeta}, \beta, \bar{\theta}, \bar{\gamma})$. ■

We can now give a proof of the main convergence result.

Proof of Theorem B.1: Convergence to a unique limit follows immediately from the assumption and Lemmas B.2, B.3 and B.4.

To prove the second statement, we first note that $\mathcal{L}$ is constant with respect to those entries of $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ for which $\bar{z}_k = 0$. It then follows from the assumption that $\frac{\partial \mathcal{L}}{\partial \theta_k} = 0$ and $\frac{\partial \mathcal{L}}{\partial \gamma_k} = 0$ for all $k = 1, \ldots, K_{\max}$ at the limit point. Similarly from Lemma B.3 we have $\frac{\partial \mathcal{L}}{\partial \zeta} = 0$ at the limit point. If $\bar{\beta} \neq \varepsilon_\beta$ we have $\frac{\partial \mathcal{L}}{\partial \beta} = 0$ at the limit point and the result follows immediately.

If $\bar{\beta} = \varepsilon_\beta$ the result can be obtained by introducing a Lagrange multiplier such that the limit point satisfies the Karush-Kuhn-Tucker conditions. ■

B Efficient Evaluation of $\omega_s(i)$, $\omega_t(i)$, $\omega_v(i)$ and $\omega_x(i)$

We derive a low-complexity computation of $\omega_s(i)$ by first inserting (B.35) into (B.39) to get

$$\omega_s(i) = \delta_{N-1}^{-1} \left(\sum_{q=\max(0,-i)}^{\min(N-1,N-1-i)} \sum_{r=0}^{q} \rho^*_{N-1+r-q} \rho_{N-1+r-q-i} - \rho_{q-r-1} \rho^*_{q+i-r-1} \right) \tag{B.52}$$

for $i = -(N-1), \ldots, N-1$. Then note that since $\mathbf{C}$ is Hermitian we have $\omega_s(-i) = \omega_s^*(i)$. We therefore restrict our attention to $i \geq 0$ in the following. We need the identity

$$\sum_{q=0}^{N-1} \sum_{r=0}^{q} z_{q,r} = \sum_{q=0}^{N-1} \sum_{k=0}^{N-1-q} z_{q+k,k}, \tag{B.53}$$

from which we get (recall that $\rho_i = 0$ for $i < 0$ and $i > N-1$)

$$\omega_s(i) = \delta_{N-1}^{-1} \sum_{q=0}^{N-1-i} (N-i-q)(\rho^*_{N-1-q} \rho_{N-1-q-i} - \rho_{q-1} \rho^*_{q-1+i}).$$

Substituting $q = N-1-\bar{q}-i$ in the first term and $q = \bar{q}+1$ in the second term we finally obtain

$$\omega_s(i) = \delta_{N-1}^{-1} \sum_{\bar{q}=0}^{N-1} c_s(\bar{q}, i) \rho_{\bar{q}} \rho^*_{\bar{q}+i}, \tag{B.54}$$

where $c_s(\bar{q}, i) \triangleq (2 - N + i + 2\bar{q})$. The above expression can be calculated as the sum of two cross-correlations in time $\mathcal{O}(N \log N)$ by using FFT techniques.

To evaluate $\omega_t(i)$ (B.40) we again insert (B.35) and get

$$\omega_t(i) = \delta_{N-1}^{-1} \left(\sum_{q=\max(0,-i)}^{\min(N-1,N-1-i)} 2\pi q \sum_{r=0}^{q} -\rho_{q-r-1}\rho_{q+i-r-1}^* + \rho_{N-1+r-q}^* \rho_{N-1+r-q-i} \right) \quad \text{(B.55)}$$

for $i = -(N-1), \ldots, N-1$. Be aware that we do not have $\omega_t(i) = \omega_t^*(-i)$. Applying (B.53), performing the same substitutions as above and following tedious, but straight-forward, algebra we finally get

$$\omega_t(i) = \frac{2\pi}{\delta_{N-1}} \sum_{\bar{q}=0}^{N-1} c_t(\bar{q}, i) \rho_{\bar{q}} \rho_{\bar{q}+i}^* \quad \text{(B.56)}$$

with

$$c_t(\bar{q}, i) \triangleq -\bar{q}(\bar{q}+i) + i\left(N - \frac{3+i}{2}\right) + \bar{q}^2 + (N-1)\left(\bar{q} - \frac{N-2}{2}\right), \quad \text{(B.57)}$$

which again can be evaluated using FFT techniques.

Omitting details, we use a similar approach to find

$$\omega_v(i) = \frac{4\pi^2}{\delta_{N-1}} \sum_{\bar{q}=0}^{N-1} c_v(\bar{q}, i) \rho_{\bar{q}} \rho_{\bar{q}+i}^* \quad \text{(B.58)}$$

$$\omega_x(i) = \frac{4\pi^2}{\delta_{N-1}} \sum_{\bar{q}=0}^{N-1} c_x(\bar{q}, i) \rho_{\bar{q}} \rho_{\bar{q}+i}^*, \quad \text{(B.59)}$$

where $\omega_v(-i) = \omega_v^*(i)$. The expression giving $\omega_v(i)$ is valid for $i \geq 0$, while that giving $\omega_x(i)$ is valid for $i = -(N-1), \ldots, N-1$. We have also defined

$$\begin{aligned} c_v(\bar{q}, i) = {} & \bar{q}(\bar{q}+i)^2 + (3\bar{q} - 2N\bar{q} - \bar{q}^2)(\bar{q}+i) \\ & + \frac{2}{3}\bar{q}^3 + (N-1)\bar{q}^2 + \left(N^2 - 3N + \frac{7}{3}\right)\bar{q} \\ & + \frac{3}{2}(i-N)^2 + \frac{1}{3}(i^3 - N^3) + \left(\frac{13}{6} - Mi\right)(i-N) + 1 \end{aligned}$$

$$\begin{aligned} c_x(\bar{q}, i) = {} & (\bar{q}^2 + 2\bar{q} - N\bar{q})(\bar{q}+i) - \frac{1}{3}\bar{q}^3 + \left(N^2 - 3N + \frac{7}{3}\right)\bar{q} - \frac{1}{6}i^3 \\ & + \left(\frac{3N^2 - 9N + 7}{6}\right) i - \frac{1}{3}N^3 + \frac{3}{2}N^2 - \frac{13}{6}N + 1. \end{aligned}$$

References

[1] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug 2005.

[2] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Process.*, vol. 40, pp. 590–600, Mar. 1992.

[3] R. Carriere and R. L. Moses, "High resolution radar target modeling using a modified Prony estimator," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 13–18, Jan. 1992.

[4] W. Bajwa, A. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, Jun. 2010.

[5] X. Andrade, J. N. Sanders, and A. Aspuru-Guzik, "Application of compressed sensing to the simulation of atomic systems," *Proc. Nat. Academy of Sciences*, vol. 109, no. 35, pp. 13 928–13 933, Jul. 2012.

[6] S.-Y. Kung, K. S. Arun, and B. D. Rao, "State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem," *J. of the Optical Soc. of America*, vol. 73, pp. 1799–1811, Dec. 1983.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[8] R. Roy and T. Kailath, "ESPRIT - estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, pp. 984–995, Jul. 1989.

[9] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2006–2021, Nov 2010.

[10] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3809–3822, Jul. 2012.

[11] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7465–7490, Nov. 2013.

[12] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, Dec. 2013.

[13] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *J. of Fourier Anal. and Applicat.*, vol. 19, no. 6, pp. 1229–1254, Dec. 2013.

[14] F. Dublanchet, J. Idier, and P. Duwaut, "Direction-of-arrival and frequency estimation using Poisson-Gaussian modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, vol. 5, Apr. 1997, pp. 3501–3504.

[15] T. L. Hansen, M.-A. Badiu, B. H. Fleury, and B. D. Rao, "A sparse Bayesian learning algorithm with dictionary parameter estimation," in *Proc. 8th IEEE Sensor Array and Multichannel Signal Process. Workshop*, Jun. 2014, pp. 385–388.

[16] M. A. Badiu, T. L. Hansen, and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Trans. Signal Process.*, vol. 65, pp. 2247–2261, May 2017.

[17] G. S. Ammar and W. B. Gragg, "The generalized Schur algorithm for the superfast solution of Toeplitz systems," in *Rational approximation and its applications in mathematics and physics*, 1987, pp. 315–330.

[18] ——, "Numerical experience with a superfast real Toeplitz solver," *Linear Algebra and its Applicat.*, vol. 121, pp. 185–206, Aug. 1989.

[19] B. R. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 5, pp. 1333–1335, Oct. 1985.

[20] I. Gohberg and I. A. Feldman, *Convolution Equations and Projection Methods for Their Solution*, ser. Translations of mathematical monographs. American Mat. Soc., 2005, vol. 41.

[21] L. Greengard and J.-Y. Lee, "Accelerating the nonuniform fast Fourier transform," *SIAM Review*, no. 46, p. 443–454, 2004.

[22] J.-Y. Lee and L. Greengard, "The nonuniform FFT of Type 3 and its applications," *J. Comput. Phys.*, no. 206, pp. 1–5, 2005.

[23] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[24] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. 9th Int. Workshop Artificial Intell. and Stat.*, Jan. 2003, pp. 3–6.

[25] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[26] J. J. Kormylo and J. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 482–488, May 1982.

[27] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. Springer, 2006.

[28] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[29] F. Dellaert, "The expectation maximization algorithm," Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20, Feb. 2002.

[30] D. Shutin, B. H. Fleury, and N. Schneckenburger, "Artifact suppression for super-resolution sparse Bayesian learning," 2018, submitted to *IEEE Trans. Signal Process.*

[31] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, "Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6257–6261, Dec. 2011.

[32] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[33] W. F. Mascarenhas, "The BFGS method with exact line searches fails for non-convex objective functions," *Math. Programming*, vol. 99, no. 1, pp. 49–61, Jan. 2004.

[34] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[35] Z. Yang and L. Xie, "On gridless sparse methods for line spectral estimation from complete and incomplete data," *IEEE Trans. Signal Process.*, vol. 63, pp. 3139–3153, Jun. 2015.

[36] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, pp. 2479–2493, Jul. 2009.

[37] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[38] M. J. D. Powell, "On search directions for minimization algorithms," *Math. Programming*, vol. 4, no. 1, pp. 193–201, Dec. 1973.

[39] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, Apr. 2000.

Paper C

A Fast Interior Point Method for Atomic Norm Soft Thresholding

Thomas L. Hansen and Tobias L. Jensen

The paper has been submitted to
Applied and Computational Harmonic Analysis

The layout has been revised.

Abstract

The atomic norm provides a generalization of the ℓ_1-norm to continuous parameter spaces. When applied as a sparse regularizer for line spectral estimation the solution can be obtained by solving a convex optimization problem. This problem is known as atomic norm soft thresholding (AST). It can be cast as a semidefinite program and solved by standard methods. In the semidefinite formulation there are $\mathcal{O}(N^2)$ dual variables and a standard primal-dual interior point method requires at least $\mathcal{O}(N^6)$ flops per iteration. That has lead researchers to consider alternating direction method of multipliers (ADMM) for the solution of AST, but this method is still somewhat slow for large problem sizes. To obtain a faster algorithm we reformulate AST as a non-symmetric conic program. That has two properties of key importance to its numerical solution: the conic formulation has only $\mathcal{O}(N)$ dual variables and the Toeplitz structure inherent to AST is preserved. Based on it we derive FastAST which is a primal-dual interior point method for solving AST. Two variants are considered with the fastest one requiring only $\mathcal{O}(N^2)$ flops per iteration. Extensive numerical experiments demonstrate that FastAST solves AST significantly faster than a state-of-the-art solver based on ADMM.

C.1 Introduction

It is well known that sparse estimation problems can be formulated as convex optimization problems using the ℓ_1-norm. The ℓ_1-norm can be generalized to continuous parameter spaces through the so-called atomic norm [1]. Convex modelling of sparsity constraints has two highly attractive traits: convex optimization problems can easily be solved both in theory [2] and in practice [3, 4], and, a number of recovery guarantees can be obtained with this framework. Such recovery guarantees are studied in signal processing under the name compressed sensing [5–7] and they generalize nicely to the atomic norm minimization approach [8–12].

The most prominent example of estimation with the atomic norm is the application to line spectral estimation [9, 11, 12], in which case it is known as atomic norm soft thresholding (AST). The popularity of AST is, partly, due to the fact that it can be cast as a semidefinite programming (SDP) problem (we refer to Sec. C.2 for a review of atomic norm minimization and AST,)

$$\begin{array}{ll} \text{minimize}_{v,x,u} & \|x-y\|_2^2 + \tau(v + w^{\mathrm{T}}u) \\ \text{subject to} & \begin{pmatrix} T(u) & x \\ x^{\mathrm{H}} & v \end{pmatrix} \succeq 0, \end{array} \tag{C.1}$$

where $v \in \mathbb{R}$, $x \in \mathbb{C}^N$, $u \in \mathbb{R}^{2N-1}$ are the variables of the problem and $y \in \mathbb{C}^N$, $\tau \in \mathbb{R}$, $w \in \mathbb{R}^{2N-1}$ are fixed (known) parameters. The function $T(u)$:

$\mathbb{R}^{2N-1} \to \mathbb{C}^{N\times N}$ outputs a complex Hermitian Toeplitz matrix constructed from u, such that the first row is $(2u_0, \ldots, u_{N-1}) + j(0, u_N, \ldots, u_{2N-2})$. To be precise, AST is obtained by selecting $w = 2e_0$ in (C.1), where e_0 is a vector with 1 in the first entry and zeros elsewhere.

The state-of-the-art method for solving (C.1) is via the alternating direction method of multipliers (ADMM) and used in [9, 13–15]. While this method is reasonably fast, it has some drawbacks. It requires the calculation of an eigenvalue decomposition in each iteration at cost $\mathcal{O}(N^3)$ floating-point operations (flops). This means that for large N it is exceedingly slow. As is often seen with proximal methods it also has slow convergence if a solution of high accuracy is requested.

Da Costa *et al.* [16] apply a low-dimensional projection of the observation vector to reduce the problem size and therefore the computational complexity of AST. In the noise-free case and under certain regularity conditions, it is shown that the estimation accuracy is not affected by doing so. However, it is clear that this approach discards observed data and the estimation accuracy will be degraded in the noisy case.

The formulation of AST in (C.1) casts it as an SDP problem. SDP problems have been subject to intensive research since the 1990's and their solution using primal-dual interior point methods (IPMs) is now understood well [2, 3, 17–19]. The Lagrangian dual of (C.1) has $\mathcal{O}(N^2)$ dual variables due to the semidefinite matrix constraint. The direct application of a standard primal-dual IPM (using direct methods for solving linear systems of equations) thus requires $\mathcal{O}(N^6)$ flops per iteration at best. Compared to this approach, proximal methods (such as ADMM) which require $\mathcal{O}(N^3)$ flops per iteration are preferable, even if they converge much slower than primal-dual interior point methods. That explains why primal-dual IPMs have not gained traction for the solution of (C.1). In this work we reformulate the constraint in (C.1) as a conic constraint on the vector $(v, x^{\mathrm{T}}, u^{\mathrm{T}})^{\mathrm{T}}$. This formulation immediately reduces the number of dual variables to $\mathcal{O}(N)$ and sets the scene for the formulation of a fast primal-dual IPM. As we will see, this approach reintroduces primal-dual IPMs as a very competitive class of algorithms for solution of the AST problem. Primal-dual IPMs have so far been rejected for the solution of AST based on the performance of general purpose solvers [9, 15] such as SeDuMi [18] and SDPT3 [19]. This work therefore also demonstrates the difference in performance that can be observed between a dedicated and a general purpose solver.

Primal-dual IPMs for conic programming typically rely on a symmetry between the primal- and dual problems. The formulation of such symmetric primal-dual IPMs require the existence of a self-scaled barrier function for the cone involved in the constraint [20, 21]. Güler [22] showed that such barrier functions exist only for the class of homogeneous and self-dual cones. The cone in our formulation is not self-dual and so a symmetric primal-dual

IPM cannot be formulated. Non-symmetric conic optimization has received some attention [23–26]. These methods generally rely on the availability of a barrier function for the dual cone and possibly evaluation of its gradient and Hessian. An easy to calculate dual barrier is not available for the cone associated to the constraint of our formulation; only an oracle which can determine membership in the dual cone is available (part of our contribution is to show how such an oracle can be constructed.)

To derive a non-symmetric primal-dual IPM which does not rely on evaluating the dual barrier or its derivatives, we formulate the augmented Karush-Kuhn-Tucker conditions and devise a dedicated approach to solving these. This approach is shown to converge to a primal-dual feasible point. A lower bound on the objective function is calculated in those iterations where a dual feasible point (as determined by the oracle) is available. From the lower bound a duality gap can be evaluated, thus providing a method for dynamically updating the barrier parameter. We show that the proposed method enjoys global convergence.

Our focus is on obtaining an algorithm which has fast runtime *in practice*, i.e., it has both low per-iteration computational complexity and it exhibits reasonably fast convergence. Theoretical statements regarding for example convergence speed are left for future work. At the core of obtaining a practically fast algorithm lies the already mentioned conic formulation (which brings the number of dual variables down to $\mathcal{O}(N)$), along with techniques for fast evaluation of linear algebra in each step of the algorithm. These evaluations are based on fast algorithms [27–30] for inversion of Toeplitz matrices. Related techniques are employed in [31–34].

We dub the algorithm FastAST. Both Newtons method and a quasi-Newton method are considered for evaluation of the search direction in FastAST. When using Newtons method the algorithm requires $\mathcal{O}(N^3)$ flops per iteration, while the quasi-Newton variant only requires $\mathcal{O}(N^2)$ flops per iteration. The numerical experiments in Section C.6 show that if a solution of medium accuracy is requested the quasi-Newton variant is faster in practice. The quasi-Newton variant is not able to obtain a solution of very high accuracy due to numerical inaccuracies in the calculation of the search direction. Solving (C.1) to high accuracy makes a difference with very large signal-to-noise ratios and in these cases the variant of FastAST using Newtons method should be used. Both the Newtons and quasi-Newton variants of FastAST are significantly faster than the ADMM-based solvers for (C.1).

Along with the primal-dual IPM presented here, we have also experimented with a primal-only version which is significantly simpler to derive. The primal-only approach does not provide a good way to select the barrier parameter (which we denote t, see Sec. C.3.2). This in turn forces the primal-only approach to use either overly conservative short-step [35] updates of the barrier parameter or it requires the barrier problem to be solved

to high accuracy for each fixed t. Both scenarios lead to a significant increase in the number of iterations required by the primal-only algorithm, resulting in significantly increased runtime. On the contrary, the primal-dual version presented in this paper allows for evaluation of a duality gap in each iteration. The duality gap gives a natural way to select the barrier parameter and also provides a very precise stopping criterion.

The paper is outlined as follows. In Section C.2 we begin with a brief review of atomic norm minimization and its application to line spectral estimation. Section C.3 details the formulation of (C.1) as a non-symmetric conic optimization program along with the theory of its solution. In Section C.4 we present a numerical algorithm for the solution of (C.1) along with implementation details. The exploitation of Toeplitz structure for fast evaluation of each step in the algorithm is discussed in Section C.5. Numerical experiments which validate the practical efficacy of the proposed algorithm are presented in Section C.6.

C.2 A Brief Review of Atomic Norm Minimization for Line Spectral Estimation

C.2.1 Line Spectral Estimation

Consider an observation vector $y \in \mathbb{C}^N$,

$$y = x + \zeta, \tag{C.2}$$

where $\zeta \in \mathbb{C}^N$ is a noise vector and $x \in \mathbb{C}^N$ is a sinusoidal signal of interest,

$$x = \sum_{k=0}^{K-1} c_k a(\omega_k), \tag{C.3}$$

where K denotes the number of sinusoids (the model order) and $\omega_k \in [0, 2\pi)$, $c_k \in \mathbb{C}$ denote the angular frequency and complex coefficient of the kth sinusoidal component. The steering vector $a(\omega)$ has entries $(a(\omega))_n = \exp(jn\omega)$ for $n = 0, \ldots, N-1$ (we use the convention $j = \sqrt{-1}$). In line spectral estimation the task is to estimate the values $(K, c_0, \ldots, c_{K-1}, \omega_0, \ldots, \omega_{K-1})$. The crux of line spectral estimation lies in obtaining estimates of the model order K and the frequencies $\{\omega_k\}$. When these are available the coefficients $\{c_k\}$ can easily be estimated using a least-squares approach. The problem is ubiquitous in signal processing; examples include direction of arrival estimation using sensor arrays [36, 37], bearing and range estimation in synthetic aperture radar [38], channel estimation in wireless communications [39] and simulation of atomic systems in molecular dynamics [40].

C.2.2 Modelling Sparsity With The Atomic Norm

The atomic norm [1, 8–10] provides a tool for describing notions of sparsity in a general setting. It is defined in terms of the *atomic set* $\mathcal{A}$. Each member of $\mathcal{A}$ is referred to as an *atom*. The atoms are the basic building block of the signal and so the atomic set defines the notion of sparsity. The atomic norm induced by $\mathcal{A}$ is defined as

$$\|x\|_{\mathcal{A}} = \inf\{\alpha > 0 : x \in \alpha \, \mathbf{conv} \, \mathcal{A}\}, \tag{C.4}$$

where $\mathbf{conv}\,\mathcal{A}$ is the convex hull of $\mathcal{A}$.

For line spectral estimation the atomic set is selected as the set of complex rotated Fourier atoms [8–10]

$$\mathcal{A} = \{a(\omega)\exp(j\phi) : \omega \in [0, 2\pi), \phi \in [0, 2\pi)\} \tag{C.5}$$

and the atomic norm can then be described as

$$\|x\|_{\mathcal{A}} = \inf_{K, \{c_k, \omega_k\}} \left\{ \sum_{k=0}^{K-1} |c_k| : x = \sum_{k=0}^{K-1} c_k a(\omega_k) \right\}. \tag{C.6}$$

It is clear that the atomic norm provides a generalization of the ℓ_1-norm to the continuous parameter space $\omega_k \in [0, 2\pi)$. Through the use of a dual polynomial characterization, the atomic norm can be expressed as the solution of an SDP,

$$\begin{aligned} \|x\|_{\mathcal{A}} = \quad & \text{minimize}_{v,u} \quad \tfrac{1}{2}\left(v + \tfrac{1}{N}\operatorname{tr} T(u)\right) \\ & \text{subject to} \quad \begin{pmatrix} T(u) & x \\ x^{\mathrm{H}} & v \end{pmatrix} \succeq 0. \end{aligned} \tag{C.7}$$

C.2.3 Atomic Norm Soft Thresholding

Atomic norm soft thresholding is obtained by generalizing the least absolute shrinkage and selection operator (LASSO) [41] to use the atomic norm instead of the ℓ_1-norm. Specifically, AST [9] solves

$$\text{minimize}_x \quad \|x - y\|_2^2 + 2\tau\|x\|_{\mathcal{A}}, \tag{C.8}$$

where $\tau > 0$ is a regularization parameter to be chosen. It is clear that AST is recovered in (C.1) by selecting $w = 2e_0$.

Estimates of the model order K and the frequencies $\{\omega_k\}$ can be obtained from the solution $(v^\star, x^\star, u^\star)$ by examining a certain dual polynomial constructed from $x^\star$. This process determines the solution in (C.6) for the recovered signal $x^\star$. Under a, somewhat restrictive, assumption concerning separation of the frequencies $\{\omega_k\}$, a number of theoretical statements can be given regarding signal and frequency recovery using AST. We refer to [8–12] for details.

We now consider the selection of the regularization parameter τ. Clearly the choice of τ crucially influences the estimation accuracy of AST. It is this parameter which determines the trade-off between fidelity and sparsity which is inherent in any estimator involving the model order K. With all else being equal, selecting larger τ gives estimates with smaller values of K. Let $\|\cdot\|_{\mathcal{A}}^*$ denote the dual norm of the atomic norm $\|\cdot\|_{\mathcal{A}}$. Then the theoretical analysis in [9] requires $\tau \geq \mathbb{E}\left[\|\zeta\|_{\mathcal{A}}^*\right]$. For a white, zero-mean circularly symmetric complex Gaussian noise vector ζ with entry-wise variance σ^2 such an upper bound is given by [9],

$$\tau = \sigma\left(1 + \frac{1}{\log(N)}\right)\sqrt{N\log(N) + N\log(4\pi\log(N))}, \tag{C.9}$$

where $\log()$ is the natural logarithm. This choice has been shown to perform well in practice and we also use it in our simulation study.

C.2.4 Reweighted Atomic Norm Minimization

The matrix $T(u)$ can be interpreted as a covariance matrix of the vector x. The trace term in (C.7) expresses a convex relaxation of the rank of $T(u)$: The rank is given by the ℓ_0 pseudo-norm (number of non-zero entries) of the vector of eigenvalues of $T(u)$. If the ℓ_0 pseudo-norm is replaced by its ℓ_1-norm convex relaxation, the trace of $T(u)$ is obtained (the trace is the sum of the eigenvalues). Assuming $T(u)$ positive-definite, the constraint in (C.7) together with the v term in expresses how plausible it is to observe x in a zero-mean circularly symmetric complex Gaussian model with covariance matrix $T(u)$. These observations leads us to intuitively view the minimization in (C.7) as an approach to find a low-rank covariance matrix $T(u)$ under which it is plausible to observe x. The value of the atomic norm then gives an indication of how well this trade-off can be achieved for a given x.

The log determinant provides a better, but non-convex, relaxation of the rank. Yang *et al.* [42] propose to replace the trace in (C.7) with a log determinant term. It is reported that this results in an increased ability to resolve closely spaced frequencies. It is known as reweighted atomic norm minimization (RAM), because it in many ways generalizes reweighted ℓ_1-norm minimization [43] to continuous parameter spaces. We now review that idea in our setting. We take the freedom to reformulate the details of the formulation in [42], but the main ideas are unchanged.

RAM is obtained by replacing $\|x\|_{\mathcal{A}}$ in (C.8) with the sparsity metric

$$\begin{aligned} \mathcal{M}_\kappa(x) = \quad & \underset{v,u}{\text{minimize}} \quad \tfrac{1}{2}\left(v + \tfrac{1}{N}\log|T(u) + \kappa I|\right) \\ & \text{subject to} \quad \begin{pmatrix} T(u) & x \\ x^{\mathrm{H}} & v \end{pmatrix} \succeq 0, \end{aligned} \tag{C.10}$$

where $\kappa > 0$ is a parameter to be chosen. The log-determinant term is concave as so the resulting problem is not convex. A majorization-minimization approach can find a locally optimal point by repeatedly solving (C.1). In the lth iteration the weighting vector w in (C.1) must be selected as

$$w_l = \frac{1}{N} T^* \left(T^{-1}(u^\star_{l-1}) \right), \tag{C.11}$$

where $u^\star_{l-1}$ is the solution obtained in the $(l-1)$th iteration. The function T^* is the adjoint[1] of the linear map T, i.e., $T^* : \mathbb{C}^{N\times N} \to \mathbb{R}^{2N-1}$ is such that $\operatorname{tr}(T(u)^{\mathrm{H}} B) = T^*(B)^{\mathrm{T}} u$ for every Hermitian $B \in \mathbb{C}^{N\times N}$. In conclusion the formulation in (C.1) directly allows for solution of both AST and RAM by appropriately selecting w. Note that a fast method to compute w_l which does not require explicit matrix inversion can be obtained using the methods described in Section C.5.

C.3 Non-symmetric Conic Optimization

C.3.1 Primal- and Dual Cones

We now return to our main focus: That of numerically solving the conic program (C.1). It can be written in the form

$$\begin{aligned} &\text{minimize} && f(\mu) \\ &\text{subject to} && \mu \in \mathcal{K}, \end{aligned} \tag{C.12}$$

where $f(\mu) = \|x - y\|_2^2 + \tau(v + w^{\mathrm{T}} u)$ and $\mathcal{K}$ is the cone defined by

$$\mathcal{K} \triangleq \left\{ \mu = \begin{pmatrix} v \\ x \\ u \end{pmatrix} : \begin{pmatrix} T(u) & x \\ x^{\mathrm{H}} & v \end{pmatrix} \succeq 0 \right\}. \tag{C.13}$$

As a precursor to deriving a primal-dual IPM, we explore the properties of $\mathcal{K}$ and its dual. It is easy to show that $\mathcal{K}$ is a proper cone (convex, closed, solid and pointed; see [17]). The dual cone $\mathcal{K}^*$ of $\mathcal{K}$ is defined as

$$\mathcal{K}^* = \{ \lambda : \langle \lambda, \mu \rangle \geq 0 \ \ \forall\, \mu \in \mathcal{K} \}. \tag{C.14}$$

Since $\mathcal{K}$ is proper, so is $\mathcal{K}^*$ [17].

[1] T^* is easy to calculate: Let B be Hermitian and let β_n denote the sum over the nth upper diagonal of B, i.e.,

$$\beta_n = \sum_{m=0}^{N-1-n} B_{m,m+n},$$

for $n = 0, \ldots, N-1$. Then $T^*(B) = (2\beta_0, 2\operatorname{Re}(\beta_1), \ldots, 2\operatorname{Re}(\beta_{N-1}), 2\operatorname{Im}(\beta_1), \ldots, 2\operatorname{Im}(\beta_{N-1}))^{\mathrm{T}}$.

In this paper (primal) variables in the cone $\mathcal{K}$ are denoted by $\mu = (v, x^{\mathrm{T}}, u^{\mathrm{T}})^{\mathrm{T}}$. The (dual) variables in the cone $\mathcal{K}^*$ are denoted by $\lambda = (\rho, s^{\mathrm{T}}, z^{\mathrm{T}})^{\mathrm{T}}$, with $\rho \in \mathbb{R}$, $s \in \mathbb{C}^N$ and $z \in \mathbb{R}^{2N-1}$. The inner product between them is defined as $\langle \lambda, \mu \rangle = \rho v + \mathrm{Re}(s^{\mathrm{H}} x) + z^{\mathrm{T}} u$.

In our proposed method, we need to check for $\lambda \in \mathcal{K}^*$. In order to characterize the dual cone $\mathcal{K}^*$, the cone of positive semidefinite Hermitian Toeplitz matrices is needed:

$$\mathcal{C} \triangleq \{u \in \mathbb{R}^{2N-1} : T(u) \succeq 0\}. \tag{C.15}$$

This cone is also proper. The corresponding dual cone $\mathcal{C}^*$ is defined analogously to (C.14). We then have the following lemma.

Lemma C.1. *The dual cone of $\mathcal{K}$ can be characterized as*

$$\mathcal{K}^* = \left\{ \lambda = \begin{pmatrix} \rho \\ s \\ z \end{pmatrix} : \left[\rho > 0, \left(z - \frac{1}{4\rho} T^*(ss^{\mathrm{H}}) \right) \in \mathcal{C}^* \right] \text{ or } [\rho = 0, s = 0, z \in \mathcal{C}^*] \right\} \tag{C.16}$$

Proof. See the appendix. □

It is clear that $\mathcal{K}$ is not self-dual ($\mathcal{K} \neq \mathcal{K}^*$) and so (C.12) is a non-symmetric conic program.

The cone $\mathcal{C}$ and its dual are defined in terms of real-valued vectors because this description simplifies the derivation of the method in Sec. C.4. These sets are, however, more naturally understood from their corresponding complex-valued forms. We therefore define the vector $u_{\mathbb{C}} = (u_0, u_1 + ju_N, u_2 + ju_{N+1}, \ldots, u_{N-1} + ju_{2N-2})^{\mathrm{T}}$ and use a similar definition of $z_{\mathbb{C}}$.

The dual cone $\mathcal{C}^*$ turns out to be the set of finite autocorrelation sequences. An excellent introduction to this set and a number of characterizations of it is given in [34] for the case of real-valued sequences. Here we extend the definition to the complex-valued case.

Definition C.1. *A vector $z_{\mathbb{C}}$ is a finite autocorrelation sequence if there exists a vector $q \in \mathbb{C}^N$ such that*[2]

$$(z_{\mathbb{C}})_k = \sum_{n=0}^{N-1-k} \bar{q}_n q_{n+k}, \qquad k = 0, \ldots, N-1. \tag{C.17}$$

In other words, z is a finite autocorrelation sequence if

$$\ldots, 0, 0, (\bar{z}_{\mathbb{C}})_{N-1}, (\bar{z}_{\mathbb{C}})_{N-2}, \ldots, (\bar{z}_{\mathbb{C}})_1, (z_{\mathbb{C}})_0, (z_{\mathbb{C}})_1, \ldots, (z_{\mathbb{C}})_{N-1}, 0, 0, \ldots \tag{C.18}$$

[2] $\bar{q}$ denotes the complex conjugate of q.

is the autocorrelation sequence of some moving average process of order $N-1$ with filter coefficients $q_1, \ldots, q_{N-1}$ and input variance $|q_0|^2$. It is well known from the theory of linear time-invariant systems, that if (C.18) is a valid autocorrelation, then it can be represented by a moving average process (i.e., there exists a coefficient vector q such that (C.17) holds).

A sequence is a valid autocorrelation sequence if and only if its Fourier transform is non-negative [44]. The Fourier transform of (C.18) is

$$Z(\omega) = z_0 + 2\sum_{k=1}^{N-1} \mathrm{Re}(z_k \exp(-j\omega k)), \qquad \omega \in [0, 2\pi). \tag{C.19}$$

Then $z \in \mathcal{C}^*$ if and only if $Z(\omega) \geq 0$ for all $\omega \in [0, 2\pi)$. The fast Fourier transform allows us to evaluate $Z(\omega)$ at a large number of points on $[0, 2\pi)$ in an efficient way. Using Lemma C.1 we therefore have a low-complexity method of (approximately) determining if $\lambda \in \mathcal{K}^*$.

We still haven't shown that the dual of the cone $\mathcal{C}$ is indeed the set of finite autocorrelation sequences. To that end, let $\tilde{\mathcal{C}}$ be the set of finite autocorrelation sequences and identify u with $u_{\mathbb{C}}$. Extending the approach of [34] to the complex-valued case, a vector u is in the dual of $\tilde{\mathcal{C}}$ if and only if $z^{\mathrm{T}}u \geq 0$ for every $z \in \tilde{\mathcal{C}}$, or, in other words, if and only if

$$z^{\mathrm{T}}u = \mathrm{Re}(z_{\mathbb{C}}^{\mathrm{H}} u_{\mathbb{C}}) = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1-k} (u_{\mathbb{C}})_k q_n \bar{q}_{n+k} = \frac{1}{2} q^{\mathrm{T}} T(u) \bar{q} \geq 0 \tag{C.20}$$

for every $q \in \mathbb{C}^N$. We can therefore identify $\mathcal{C}$ with $\tilde{\mathcal{C}}^*$. Since $\tilde{\mathcal{C}}$ is a proper cone, we have $\mathcal{C}^* = \tilde{\mathcal{C}}^{**} = \tilde{\mathcal{C}}$.

C.3.2 Barrier Functions

Interior point methods are build on the idea of a barrier function $F : \mathbf{int}\,\mathcal{K} \to \mathbb{R}$ associated to the cone $\mathcal{K}$ ($\mathbf{int}\,\mathcal{K}$ denotes the interior of $\mathcal{K}$). The barrier function must be a smooth and strongly convex function with $F(\mu_k) \to \infty$ for every sequence of points $\mu_k \in \mathbf{int}\,\mathcal{K}$ with limit point $\tilde{\mu} \in \mathbf{bd}\,\mathcal{K}$, where $\mathbf{bd}\,\mathcal{K}$ denotes the boundary of $\mathcal{K}$. The typical approach to IPMs also assumes that the barrier function is logarithmically homogeneous (LH). F is a LH barrier function for the cone $\mathcal{K}$ if there exists a $\theta_F > 0$ such that $F(\alpha\mu) = F(\mu) - \theta_F \log(\alpha)$ for all $\alpha > 0$, $\mu \in \mathbf{int}\,\mathcal{K}$. The value θ_F is called the degree of the barrier.

We will use the following well-known properties of a LH barrier function F for $\mathcal{K}$ [17, 20, 21]: If $\mu \in \mathbf{int}\,\mathcal{K}$, then

$$\left\langle -\nabla_\mu F(\mu), \mu \right\rangle = \theta_F, \tag{C.21}$$

$$-\nabla_\mu F(\mu) \in \mathbf{int}\,\mathcal{K}^*. \tag{C.22}$$

The usefulness of barrier functions is clear when considering their role in path-following methods. A primal-only path-following method finds a solution to (C.12) by iteratively solving

$$\begin{array}{ll} \text{minimize} & f(\mu) + t^{-1}F(\mu) \\ \text{subject to} & \mu \in \mathbf{int}\,\mathcal{K} \end{array} \tag{C.23}$$

for an increasing sequence of values $t > 0$. In each step μ is initialized with the solution of the previous step. This approach is desirable because each step can be solved by an algorithm for unconstrained optimization such as Newtons method.

In this paper we use the standard log-determinant barrier function for $\mathcal{K}$:

$$\begin{aligned} F(\mu) &= -\log \left| \begin{pmatrix} T(u) & x \\ x^{\mathrm{H}} & v \end{pmatrix} \right| \\ &= -\log|T(u)| - \log(v - x^{\mathrm{H}}T^{-1}(u)x), \qquad \text{for } \mu \in \mathbf{int}\,\mathcal{K}. \end{aligned} \tag{C.24}$$

It is easy to show that it is LH with degree $\theta_F = N + 1$.

C.3.3 Solvability

We now consider conditions for the problem (C.1) to be solvable. An optimization problem is solvable when a feasible point exists and its objective is bounded below on the feasible set.

Lemma C.2. *The function $f(\mu)$ is bounded below on $\mu \in \mathcal{K}$ if and only if $\tau \geq 0$ and $w \in \mathcal{C}^*$.*

Proof. If $\tau < 0$ or $w \notin \mathcal{C}^*$ there exists $\mu \in \mathcal{K}$ with $x = 0$ such that $\tau v + \tau w^{\mathrm{T}}u < 0$. Note that $\alpha\mu \in \mathcal{K}$ for any $\alpha \geq 0$ and that $\lim_{\alpha\to\infty} f(\alpha\mu) = -\infty$, so $f(\mu)$ is unbounded below on $\mu \in \mathcal{K}$.

Conversely, if $\tau \geq 0$ and $w \in \mathcal{C}^*$, we have $\tau v \geq 0$ and $\tau w^{\mathrm{T}}u \geq 0$ for every $\mu \in \mathcal{K}$. So $f(\mu) \geq 0$ for $\mu \in \mathcal{K}$. □

Since a primal feasible point always exists (take for example $v = 1, x = 0, u = e_0$), the problem (C.1) is solvable if and only if the conditions in Lemma C.2 are fulfilled. These conditions can easily be checked prior to executing the algorithm and we assume that the problem is solvable in the following.

C.3.4 Optimality Conditions

With the conic modelling machinery in place, we can begin to analyze the solution of (C.1) by considering the non-symmetric conic formulation (C.12). The Lagrangian is

$$L(\mu, \lambda) = \|x - y\|_2^2 + \tau(v + w^{\mathrm{T}}u) - \langle \lambda, \mu \rangle \tag{C.25}$$

and the dual of the conic program (C.12) is

$$\begin{array}{ll} \text{maximize} & -\frac{1}{4}\|s\|_2^2 - \mathrm{Re}(y^{\mathrm{H}}s) \\ \text{subject to} & \lambda \in \mathcal{K}^*,\ \rho = \tau,\ z = \tau w. \end{array} \tag{C.26}$$

Notice that by taking the dual of (C.12) instead of (C.1), the number of dual variables is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ (see [9] for an explicit formulation of the dual of (C.1)). This is the reason why, from a computational point of view, it is beneficial to work with the form (C.12) instead of (C.1), as is usually done in the literature.

Since f is convex, the Karush-Kuhn-Tucker (KKT) are necessary and sufficient [17, Sec. 5.9] for variables $(\mu^\star, \lambda^\star)$ to be solutions of the primal and dual problems (C.12) and (C.26). The KKT conditions are

$$\left\{ \begin{array}{l} \nabla_\mu L(\mu^\star, \lambda^\star) = 0 \\ \mu^\star \in \mathcal{K} \\ \lambda^\star \in \mathcal{K}^* \\ \langle \lambda^\star, \mu^\star \rangle = 0 \end{array} \right\}, \tag{C.27}$$

where the gradient operator is defined as $\nabla_\mu f = (\nabla_v f, \nabla_x f^{\mathrm{T}}, \nabla_u f^{\mathrm{T}})^{\mathrm{T}}$. The gradient with respect to the complex vector $x = a + jb$ is to be understood as[3] $\nabla_x f = \nabla_a f + j\nabla_b f$.

Instead of directly solving the KKT conditions, our primal-dual interior point algorithm finds solutions $(\mu^{(t)}, \lambda^{(t)})$ of the augmented KKT conditions [17, 45]

$$\left\{ \begin{array}{l} \nabla_\mu L\left(\mu^{(t)}, \lambda^{(t)}\right) = 0 \\ \mu^{(t)} \in \mathbf{int}\,\mathcal{K} \\ \lambda^{(t)} \in \mathbf{int}\,\mathcal{K}^* \\ \lambda^{(t)} = -t^{-1}\nabla_\mu F\left(\mu^{(t)}\right) \end{array} \right\} \tag{C.28}$$

for an increasing sequence of values $t > 0$. It is easy to realize that $\left(\mu^{(t)}, \lambda^{(t)}\right)$ solves (C.28) only if $\mu^{(t)}$ is a solution of the barrier problem (C.23). This observation provides the link between primal-only barrier methods and primal-dual interior point methods. The set of values $\left\{ \left(\mu^{(t)}, \lambda^{(t)}\right) : t > 0 \right\}$ is known as the primal-dual central path. The primal-dual central path converges to the desired solution in the sense that $\lim_{t\to\infty} \left(\mu^{(t)}, \lambda^{(t)}\right) = (\mu^\star, \lambda^\star)$ [17, 45].

The last condition in (C.28) is known as the augmented complementary slackness condition. It follows from (C.22) that the second and fourth condition in (C.28) together imply $\lambda^{(t)} \in \mathbf{int}\,\mathcal{K}^*$, so the third condition can be dropped.

[3] This definition is actually twice the Wirtinger derivative of f with respect to $\bar{x}$.

From (C.21) it follows that the duality gap for the primal-dual problems (C.12) and (C.26) at a point on the primal-dual central path is $\left\langle \lambda^{(t)}, \mu^{(t)} \right\rangle = \theta_F / t$ [17, 20]. So solving the augmented KKT gives a primal feasible solution $\mu^{(t)}$ which is no more than $(N+1)/t$ suboptimal. Consequently, an arbitrarily accurate solution can be obtained by solving (C.28) for sufficiently large t.

C.3.5 Obtaining a Solution of the Augmented KKT Conditions

We now define $v^{(t)}, x^{(t)}, u^{(t)}, \rho^{(t)}, s^{(t)}$ and $z^{(t)}$ as the entries of $\mu^{(t)}$ and $\lambda^{(t)}$. By solving the first condition in (C.28) (the stationarity condition), we get

$$\rho^{(t)} = \tau, \quad z^{(t)} = \tau w, \quad s^{(t)} = 2(x^{(t)} - y). \tag{C.29}$$

We continue by writing out the last condition of (C.28) which is known as the augmented complementary slackness condition. Solving for $v^{(t)}$ and $x^{(t)}$ and inserting the relations above, we get

$$v^{(t)} = (\tau t)^{-1} + \left(x^{(t)}\right)^{\mathrm{H}} T^{-1}\left(u^{(t)}\right) x^{(t)} \tag{C.30}$$

$$x^{(t)} = T\left(u^{(t)}\right) T^{-1}\left(u^{(t)} + 2^{-1}\tau e_0\right) y. \tag{C.31}$$

Finally, solving $z^{(t)} = -t^{-1}\nabla_u F\left(\mu^{(t)}\right)$ for $u^{(t)}$ and inserting the above gives

$$\tau w - \tau T^*\left(\phi\phi^{\mathrm{H}}\right) - t^{-1} T^*\left(T^{-1}(u^{(t)})\right) = 0, \tag{C.32}$$

where $\phi = T^{-1}\left(u^{(t)} + 2^{-1}\tau e_0\right) y$.

For a given $t > 0$ we can obtain the corresponding point on the primal-dual central path as follows: First a solution $u^{(t)}$ of (C.32) that fulfills $u^{(t)} \in \mathbf{int}\,\mathcal{C}$ is found (existence of such a solution is shown below). Then the point $\left(\mu^{(t)}, \lambda^{(t)}\right)$ is obtained by inserting into (C.29), (C.30) and (C.31). It is easy to show from $u^{(t)} \in \mathbf{int}\,\mathcal{C}$ that $\mu^{(t)} \in \mathbf{int}\,\mathcal{K}$ and so $\left(\mu^{(t)}, \lambda^{(t)}\right)$ solves (C.28) and it is a primal-dual central point.

How can we obtain a solution $u^{(t)} \in \mathbf{int}\,\mathcal{C}$ of (C.32)? The left-hand side of (C.32) is recognized as the gradient of $h_t(u) = g(u) + t^{-1}G(u)$, with

$$g(u) = \tau w^{\mathrm{T}} u + \tau y^{\mathrm{H}} T^{-1}\left(u + 2^{-1}\tau e_0\right) y \tag{C.33}$$

$$G(u) = -\log|T(u)|. \tag{C.34}$$

Now consider the barrier problem

$$\begin{aligned} &\text{minimize} && h_t(u) \\ &\text{subject to} && u \in \mathbf{int}\,\mathcal{C}. \end{aligned} \tag{C.35}$$

The gradient of h_t vanishes at the solution of (C.35) because G is a LH barrier function for $\mathcal{C}$. So solving (C.32) with $u^{(t)} \in \mathbf{int}\,\mathcal{C}$ is equivalent to solving (C.35). The problem (C.35) is always solvable if the problem (C.1) is solvable (thus proving that there exists a $u^{(t)} \in \mathbf{int}\,\mathcal{C}$ that solves (C.32) if (C.1) is solvable).

The idea of the primal-dual IPM presented in the following section is to use an iterative algorithm for unconstrained optimization (either Newtons method or a quasi-Newton method) to solve (C.35). However, we do not need to exactly solve (C.35) for a sequence of values $t > 0$. In each iteration of the solver the value of t can be updated in a dynamic manner based on the duality gap.

C.4 The Primal-Dual Interior Point Method

We now propose a primal-dual IPM for the solution of (C.12). Let (μ_i, λ_i, t_i) denote (μ, λ, t) in iteration i. The proposed method is given in Algorithm C.1. In the remainder of this section, each step of the algorithm is discussed in detail.

Low-complexity evaluation of the steps in the above algorithm are presented in Sec. C.5. With these approaches, the computational complexity is dominated by the evaluation of the search direction. For this step we propose to use either Newton's method or a quasi-Newton method. The quasi-Newton method has much lower computational complexity per iteration and is also faster in practice. It is, however, not able to obtain a solution of high accuracy. If a highly accurate solution is required, Newtons method is preferred. We refer to the numerical evaluation in Sec. C.6 for a detailed discussion thereof.

C.4.1 Determining the Search Direction Using Newtons Method

Applying Newtons method to solve (C.35) we get the search direction

$$\Delta u = -\left(\nabla_u^2 h_{t_i}(u_{i-1})\right)^{-1} \nabla_u h_{t_i}(u_{i-1}), \tag{C.36}$$

where $\nabla_u^2 h_{t_i}(u_{i-1})$ denotes the Hessian of h_{t_i} evaluated at u_{i-1}. As discussed in Sec. C.5, the Hessian can be evaluated in $\mathcal{O}(N^3)$ flops and the same cost is required for solution of the system (C.36).

Algorithm C.1: Primal-dual interior point method for fast atomic norm soft thresholding (FastAST).

Parameters: $\gamma > 1$.
Input: Initial values $u_0 \in \mathcal{C}$ and $t_1 > 0$.
Set objective lower bound $f_{\text{LB}} = -\infty$.
for $i = 1, 2, \ldots$ **do**
  Determine the search direction Δu.
  Perform a line search along Δu to obtain the step size α.
  Update estimate $u_i = u_{i-1} + \alpha \Delta u$.
  Form primal-dual variables (μ_i, λ_i) using (C.29), (C.30) and (C.31).
  if $\lambda_i \in \mathcal{K}^*$ **then**
    Update lower bound on objective
    $f_{\text{LB}} = \max\Big(f_{\text{LB}}, -\frac{1}{4}\left\|s_i\right\|_2^2 - \text{Re}(y^{\text{H}} s_i)\Big)$.
  end
  Evaluate duality gap $\eta_i = f(\mu_i) - f_{\text{LB}}$.
  Terminate if the stopping criterion is satisfied.
  Update barrier parameter $t_{i+1} = \max\Big(t_i, \gamma \frac{N+1}{\eta_i}\Big)$.
end
Output: Primal-dual solution (μ_i, λ_i).

C.4.2 Determining the Search Direction Using L-BFGS

In scenarios with large N the computation time for evaluation of the Newton search direction can become prohibitively large. In these cases we propose to use the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [46] for the solution of (C.35). L-BFGS enjoys several properties which are instrumental in obtaining an algorithm that has low computational requirements in each iteration:

- L-BFGS uses only gradient information and the gradient of g and G can be evaluated with low computational complexity,[4] see Section C.5.
- We show that by appropriately modifying the L-BFGS two-loop recursion, it can be used for the solution of (C.35) in a computational efficient manner. Each iteration only requires a single computation of the gradients of g (C.33) and G (C.34), combined with simple level 1 Basic Linear Algebra Subroutine (BLAS) operation even though t is increased in every iteration. It is this property, and not the limited memory requirements, that makes L-BFGS preferable over other quasi-Newton methods (such as vanilla BFGS) for our purposes.

[4] To speed up convergence, our implementation also uses an approximation of the diagonal of the Hessian of h_{t_i}.

Since $t_i \neq t_{i-1} \neq \ldots$, the normal formulation of L-BFGS does not apply. We now demonstrate that a simple modification of the L-BFGS two-loop recursion [46] overcomes this limitation. At the end of the ith iteration, the following difference vectors are calculated and saved for use in later iterations:

$$r_i = u_i - u_{i-1} \tag{C.37}$$

$$q_i = \nabla_u g(u_i) - \nabla_u g(u_{i-1}) \tag{C.38}$$

$$Q_i = \nabla_u G(u_i) - \nabla_u G(u_{i-1}). \tag{C.39}$$

This set of vectors is retained for M iterations. The two-loop recursion in Algorithm C.2 can then be use to calculate the search direction Δu. This algorithm calculates the normal L-BFGS search direction for minimization of h_{t_i}, as if $t_i = t_{i-1} = \ldots$. That can be achieved because L-BFGS only depends on t_i through the quantities $\nabla_u h_{t_i}(u_k) = \nabla_u g(u_k) + t_i^{-1} \nabla_u G(u_k)$, for $k = i-1, \ldots, i-M-1$. The gradients $\nabla_u g(u_k)$ and $\nabla_u G(u_k)$ need only be calculated once to allow $\nabla_u h_{t_i}(u_k)$ to be calculated for the current value of t_i.

Algorithm C.2: Calculating the search direction based on the L-BFGS two-loop recursion.

Parameters: Number of saved difference vectors M.
Input: Current iteration number i and parameter t_i. Saved difference vectors r_k, q_k, Q_k for $k = i-1, i-2, \ldots, \max(i-M, 1)$. Current gradient vector $\nabla_u h_{t_i}(u_{i-1})$ and initial Hessian approximation $\hat{H}_i$.

$d \leftarrow -\nabla_u h_{t_i}(u_{i-1})$
for $k = i-1, i-2, \ldots, \max(i-M, 1)$ **do**
  $\psi_k \leftarrow q_k + t_i^{-1} Q_k$
  $\sigma_k \leftarrow \frac{r_k^{\mathrm{T}} d}{r_k^{\mathrm{T}} \psi_k}$
  $d \leftarrow d - \sigma_k \psi_k$
end
$d \leftarrow \hat{H}_i^{-1} d$
for $k = \max(i-M, 1), \max(i-M, 1)+1, \ldots, i-1$ **do**
  $\beta_k \leftarrow \frac{\psi_k^{\mathrm{T}} d}{\psi_k^{\mathrm{T}} r_k}$
  $d \leftarrow d + r_k (\sigma_k - \beta_k)$
end

Output: Search direction $\Delta u = d$

In each iteration the "initial" Hessian $\hat{H}_i$ should be chosen as an approximation of the Hessian of h_{t_i} evaluated at u_{i-1}. It is the matrix upon which

L-BFGS successively applies rank-2 updates to form the Hessian approximation, which is used for calculating the search direction. An easy, and popular, choice for the initial Hessian is the identity matrix $\hat{H}_i = I$. Through numerical experiments we have seen that this choice leads to slow convergence. It turns out that the slow convergence is caused by the scaling of the Hessian, leading to non-acceptance of a full Newton step (i.e., α is selected much smaller than 1). Using a diagonal approximation of the true Hessian remedies this, but, unfortunately, it cannot be calculated with low computational complexity. (Our best attempt at devising a fast evaluation of the Hessian diagonal yielded cubic complexity $\mathcal{O}(N^3)$, the same as evaluation of the full Hessian.) Instead our algorithm uses the following heuristic approximation of the diagonal Hessian

$$\hat{H}_i = \operatorname{diag}\left(1, \frac{N-1}{2N}, \ldots, \frac{1}{2N}, \frac{N-1}{2N}, \ldots, \frac{1}{2N}\right) \left(\nabla_u^2 h_{t_i}(u_{i-1})\right)_{0,0}, \quad \text{(C.40)}$$

where $\left(\nabla_u^2 h_{t_i}(u_{i-1})\right)_{0,0}$ is the $(0,0)$th entry of the true Hessian evaluated at u_{i-1}. This approximation can be calculated with low computational complexity as demonstrated in Section C.5. The approximation is motivated as follows: The diagonal entries are scaled according to the number of times the corresponding entry of u appears in $T(u)$. This scaling resembles that in the biased autocorrelation estimate (except for a factor of 2 caused by the scaling of the diagonal in the definition of $T(u)$). In our numerical experiments, we have observed the above approximation to be fairly accurate; each entry typically takes a value within $\pm 50\,\%$ of the true value. To this end we note that only a crude approximation is needed, since the role of $\hat{H}_i$ is to account for the scaling of the problem. Our numerical investigation suggests that using the approximation (C.40) leads to only marginally slower convergence, compared to using a diagonal Hessian approximation using the diagonal of the true Hessian.

A final note on our adaptation of L-BFGS is that the usual observations regarding positive definiteness of the approximated Hessian remain valid. First note that the objective upon which L-BFGS is applied (h_{t_i}) is a strictly convex function for $u \in \mathbf{int}\,\mathcal{C}$. It follows that the initial Hessian approximation $\hat{H}_i$ is positive definite. Also, the curvature condition $r_k^{\mathrm{T}} \psi_k > 0$ is valid for all k. Then the approximated Hessian is positive definite and the calculated search direction Δu is a descent direction [4, 46].

C.4.3 Line Search

The line search along the search direction Δu is a simple backtracking line search starting at $\alpha = 1$. A step size is accepted if the new point is strictly feasible, i.e., if $u_{i-1} + \alpha \Delta u \in \mathbf{int}\,\mathcal{C}$. It is then easy to show that the primal

solution μ_i calculated from inserting u_i into (C.30) and (C.31) is strictly primal feasible ($\mu_i \in \mathbf{int}\,\mathcal{K}$).

To guarantee that the objective is sufficiently decreased, the Armijo rule is also required for acceptance of a step size α:

$$h_{t_i}(u_{i-1} + \alpha\Delta u) \leq h_{t_i}(u_{i-1}) + c\alpha\Delta u^{\mathrm{T}}\nabla_u h_{t_i}(u_{i-1}), \tag{C.41}$$

where $0 < c < 1$ is a suitably chosen constant.

C.4.4 The Duality Gap and Update of t

The line search guarantees that the primal solution is strictly feasible in all iterations, i.e., that $\mu_i \in \mathbf{int}\,\mathcal{K}$. Dual feasibility of a solution λ_i obtained from (C.29) is not guaranteed. The algorithm therefore checks for $\lambda_i \in \mathcal{K}^*$ using the approximate approach described in Sec. C.3.1.

Let $f^\star$ denote the optimal value of the problem (C.12). If λ_i is dual feasible, the objective of the dual (C.26) provides a lower bound on the optimal value, i.e.,

$$f^\star \geq -\frac{1}{4}\left\|s_i\right\|_2^2 - \mathrm{Re}(y^{\mathrm{H}}s_i). \tag{C.42}$$

The algorithm always retains the largest lower bound it has encountered in f_{LB}. From the lower bound, a duality gap η_i can be evaluated in each iteration:

$$\eta_i = f(\mu_i) - f_{\mathrm{LB}}. \tag{C.43}$$

This value gives an upper bound on the sub optimality of the solution μ_i, i.e., $f(\mu_i) - f^\star \leq \eta_i$.

Recall that the algorithm is "aiming" for a solution of the augmented KKT conditions (C.28). At this solution, the duality gap is θ_F/t_{i+1}. The next value of t can then be determined so that the algorithm is aiming for a suitable (not too large, not too small) decrease in the duality gap, i.e., we select t_{i+1} such that $\eta_i/\gamma = \theta_F/t_{i+1}$ for some preselected $\gamma > 1$. Recall that the degree of the barrier F is $\theta_F = N + 1$. To guarantee convergence it is also imposed that t_i is a non-decreasing sequence.

C.4.5 Termination

The duality gap provides a natural stopping criterion. The proposed algorithm terminates based on either the duality gap ($\eta_i < \varepsilon_{\mathrm{abs}}$) or the relative duality gap ($\eta_i/f(\mu_i) < \varepsilon_{\mathrm{rel}}$). The relative duality gap is a sensible stopping criterion because $f(\mu) \geq 0$ as is seen in the proof of Lemma C.2.

Algorithm C.1 is guaranteed to terminate at a point that fulfills either of the two stopping criteria listed above. To see why that is the case, consider

a scenario where t_i converges to some finite constant $\tilde{t}$ as $i \to \infty$. Then, as $i \to \infty$, the algorithm implements L-BFGS with a backtracking line search to minimize $h_{\tilde{t}}$. Thus u_i converges to the minimizer $u^{(\tilde{t})}$ of $h_{\tilde{t}}$. Let $(\mu^{(\tilde{t})}, \lambda^{(\tilde{t})})$ denote the corresponding primal and dual variables calculated from (C.30), (C.31) and (C.29).

Now, $(\mu^{(\tilde{t})}, \lambda^{(\tilde{t})})$ constitute a solution to (C.28) with $t = \tilde{t}$. Then $\lambda^{(\tilde{t})} \in \mathbf{int}\, \mathcal{K}^*$ follows from (C.22). Further, we have from (C.43), (C.29) and (C.21) that the duality gap η_i converges to $\left\langle \mu^{(\tilde{t})}, \lambda^{(\tilde{t})} \right\rangle = \theta_F / \tilde{t}$ as $i \to \infty$. However, that implies $t_{i+1} = \gamma \theta_F / \eta_i = \gamma \tilde{t} > \tilde{t}$ in the limit, a contradiction to the assumption that t_i converges to $\tilde{t}$. This means that t_i does not converge to a finite value and, as it is non-decreasing, it must diverge to $+\infty$. It is also evident that the duality gap $\eta_i \to 0$ as $t_i \to \infty$, and so either of the stopping criteria are eventually fulfilled.

C.4.6 Initialization

The algorithm must be initialized with primal variable $u_0 \in \mathcal{C}$ and barrier parameter $t_1 > 0$. To determine a suitable value of the initial barrier parameter t_1 we first identify a primal-dual feasible point from which the duality gap can be evaluated. A primal-dual feasible point can be obtained by assuming[5] $w \in \mathbf{int}\, \mathcal{C}^*$ and iterating these steps:

1. Set $u = (10\|y\|_2^2)/N, 0, \ldots, 0)^{\mathrm{T}}$.
2. Calculate (μ, λ) from u based on (C.29), (C.30) and (C.31).
3. If $\lambda \in \mathcal{K}^*$, terminate, otherwise double the first entry of u and go to step 2.

The value of u in Step 1 has been chosen heuristically. It is easy to see that u stays primal feasible throughout. The above scheme finds a primal-dual feasible point (μ, λ). The corresponding duality gap is $\eta_0 = \langle \mu, \lambda \rangle$. We then select $t_1 = \gamma \theta_F / \eta_0$. The corresponding value of u is used as the initial value of the primal variable u_0.

The above scheme is guaranteed to find a primal-dual feasible point because $u \to (\infty, 0, \ldots, 0)^{\mathrm{T}}$. Then, following (C.31), we have $x \to y$ and so $s \to 0$. Considering the result in Lemma C.1 and the assumption $w \in \mathbf{int}\, \mathcal{C}^*$, we get that λ converges to a point $\tilde{\lambda} \in \mathbf{int}\, \mathcal{K}^*$.

[5]The problem (C.1) is solvable if and only if $w \in \mathcal{C}^*$. The restriction to the interior has no practical effect.

C.5 Fast Computations

For brevity, iteration indices are dropped in the following. The computationally demanding steps of Algorithm C.1 all involve the determinant or the inverse of Toeplitz matrices $T(u)$ and $T(u + 2^{-1}\tau e_0)$. In this section we demonstrate how the Toeplitz structure can be exploited to significantly reduce the computationally complexity of these evaluations. The exploitation of such structure for fast solution of optimization problems have previously been seen [31, 32], including for evaluation of the gradient and Hessian of the barrier function G [33, 34].

C.5.1 Fast Algorithms for Factorizing a Toeplitz Inverse

Our computational approach is based on the following factorizations of Toeplitz inverses. The Gohberg-Semencul formula [27, 47] gives a factorization of the inverse of a Toeplitz matrix $T(u)$,

$$T^{-1}(u) = \delta_{N-1}^{-1}(U^{\mathrm{H}}U - VV^{\mathrm{H}}), \tag{C.44}$$

where the entries of Toeplitz matrices U and V are

$$U_{n,m} = \rho_{N-1+n-m}, \tag{C.45}$$

$$V_{n,m} = \rho_{n-m-1}, \tag{C.46}$$

for $n, m = 0, \ldots, N-1$. Note that $\rho_n = 0$ for $n < 0$ and $n > N-1$; thus U is unit upper triangular ($\rho_{N-1} = 1$) and V is strictly lower triangular.

The values δ_n and ρ_n for $n = 0, \ldots, N-1$ can be computed with a generalized Schur algorithm in $\mathcal{O}(N \log^2 N)$ flops [27]. Alternatively, the Levinson-Durbin algorithm can be used to obtain the decomposition in $\mathcal{O}(N^2)$ flops. The latter algorithm is significantly simpler to implement and is faster for small N. In [28] it is concluded that the Levinson-Durbin algorithm requires fewer total operations than the generalized Schur algorithm for $N \leq 256$.

We will also use a Cholesky factorization of $T^{-1}(u)$, namely

$$T^{-1}(u) = PDP^{\mathrm{H}} \tag{C.47}$$

where P is unit upper triangular and D is diagonal. The diagonal matrix $D = \mathrm{diag}(\delta_0^{-1}, \ldots, \delta_{N-1}^{-1})$ is inherently computed when the generalized Schur algorithm is executed [27]. The generalized Schur algorithm does not compute the matrix P. The Levinson-Durbin algorithm inherently computes both P and D, a property which we exploit for evaluation of the Hessian of the barrier function G.

In the following we let $\rho_0, \ldots, \rho_{N-1}$ and $\delta_0, \ldots, \delta_{N-1}$ be the entries obtained by executing the generalized Schur or Levinson-Durbin algorithm with either $T^{-1}(u)$ or $T^{-1}(u + 2^{-1}\tau e_0)$; which one will be clear from the context.

C.5.2 Evaluating the Objective and the Primal Variables

We first discuss evaluation of the objective $h_t(u) = g(u) + t^{-1}G(u)$. Since P in (C.47) has unit diagonals, it is easy to obtain

$$G(u) = -\log|T(u)| = -\sum_{n=0}^{N-1} \log \delta_n. \tag{C.48}$$

To evaluate $g(u)$ insert (C.44) into (C.33) and realize that all matrix-vector products involve Toeplitz matrices. Vector multiplication onto a Toeplitz matrix can be performed using the fast Fourier transform (FFT) in $\mathcal{O}(N \log N)$ flops (such products are convolutions, see e.g. [34] for details). In conclusion, the dominant cost of evaluating $h_t(u)$ is the execution of the generalized Schur (or Levinson-Durbin) algorithm.

Evaluating the primal variables $v^{(t)}$ and $x^{(t)}$ in (C.30)–(C.31) similarly amounts to vector products onto Toeplitz matrices and their inverses.

The line search in Algorithm C.1 must check for $u \in \mathcal{C}$, i.e., if $T(u) \succ 0$. The generalized Schur (or Levinson-Durbin) algorithm can again be used here, as $T(u) \succ 0$ if and only if $\delta_n > 0$ for $n = 0, \ldots, N-1$.

C.5.3 Evaluating the Gradients

The following gradients must be evaluated in each iteration of Algorithm C.1:

$$\nabla_u g(u) = \tau w - \tau T^*\left(\phi\phi^{\mathrm{H}}\right) \tag{C.49}$$

$$\nabla_u G(u) = -T^*\left(T^{-1}(u)\right). \tag{C.50}$$

We first consider the term $T^*(\phi\phi^{\mathrm{H}})$. The vector ϕ can be evaluated with low complexity (confer the evaluation of primal variables, above). Let $\beta_n \in \mathbb{C}$ denote the sum over the nth upper diagonal of $\phi\phi^{\mathrm{H}}$ for $n = 0, \ldots, N-1$, i.e.,

$$\beta_n = \sum_{m=0}^{N-1-n} (\phi\phi^{\mathrm{H}})_{m,m+n} = \sum_{m=0}^{N-1-n} \phi_m \bar{\phi}_{m+n}. \tag{C.51}$$

It is recognized that the values $\beta_0, \ldots, \beta_{N-1}$ can be calculated as a correlation, which can be implemented using FFTs in $\mathcal{O}(N \log N)$ flops. Then $T^*(\phi\phi^{\mathrm{H}})$ can be obtained by concatenating (and scaling) the real and imaginary parts of β,

$$T^*(\phi\phi^{\mathrm{H}}) = (2\beta_0, 2\,\mathrm{Re}(\beta_1), \ldots, 2\,\mathrm{Re}(\beta_{N-1}), 2\,\mathrm{Im}(\beta_1), \ldots, 2\,\mathrm{Im}(\beta_{N-1}))^{\mathrm{T}}. \tag{C.52}$$

Now consider evaluation of the term $T^*\left(T^{-1}(u)\right)$. Let $\rho_0, \ldots, \rho_{N-1}$ and δ_{N-1} denote the entries in the decomposition (C.44) obtained by applying

the generalized Schur (or Levinson-Durbin) algorithm to $T^{-1}(u)$. The sum over the diagonals of $T^{-1}(u)$ can then be rewritten as

$$\tilde{\beta}_n = \sum_{m=0}^{N-1-n} (T^{-1}(u))_{m,m+n} \tag{C.53}$$

$$= \delta_{N-1}^{-1} \sum_{k=0}^{N-1} (n - N + 2(k+1)) \rho_k \bar{\rho}_{k+n}\,, \tag{C.54}$$

see [31, 32] for details. The above is recognized as two correlations, thus allowing low-complexity evaluation. The vector $T^*\big(T^{-1}(u)\big)$ is found by scaling and concatenating the real and imaginary parts of $\tilde{\beta}$, analogously to (C.52).

C.5.4 Evaluating the Full Hessian

When Newtons method is used to determine the search direction, the Hessian of h_t must be evaluated. We now derive an approach to calculate the Hessians of g and G, from which the required Hessian is easily found.

The (n, m)th entry of the Hessian of g is

$$\Big(\nabla_u^2 g(u)\Big)_{n,m} = 2\tau\phi^{\mathrm{H}}(E_n + E_n^{\mathrm{H}})T^{-1}(u + 2^{-1}\tau e_0)(E_m + E_m^{\mathrm{H}})\phi, \tag{C.55}$$

where

$$E_n = \begin{cases} I & n = 0 \\ \tilde{E}^n & 1 \leq n \leq N-1 \\ -jE_{n-N+1} & N \leq n \leq 2N-1. \end{cases} \tag{C.56}$$

The matrix $\tilde{E}$ is the lower shift matrix, i.e., it is has ones on the lower subdiagonal and zeros elsewhere. This means that $T(e_n) = E_n + E_n^{\mathrm{H}}$. The mth column of the Hessian is then

$$\Big(\nabla_u^2 g(u)\Big)_m = \tau T^*\Big(d_m\phi^{\mathrm{H}} + \phi d_m^{\mathrm{H}}\Big)\,, \tag{C.57}$$

where we let d_m denote a vector $d_m = T^{-1}(u + 2^{-1}\tau e_0)(E_m + E_m^{\mathrm{H}})\phi$. A column of the Hessian can then be calculated in $\mathcal{O}(N \log N)$ flops by using (C.44) and writing out the sum over the upper diagonals of $d_m\phi^{\mathrm{H}}$ (similarly to (C.51)). The full Hessian of g is then obtained in $\mathcal{O}(N^2 \log N)$ flops.

To evaluate the Hessian of the barrier function G we generalize the approach of [34] to the complex-valued case. The (n, m)th entry of the Hessian is

$$\Big(\nabla_u^2 G(u)\Big)_{n,m} = \mathrm{tr}\Big(T^{-1}(u)(E_n + E_n^{\mathrm{H}})T^{-1}(u)(E_m + E_m^{\mathrm{H}})\Big) \tag{C.58}$$

$$= \mathrm{Re}\Big(\mathrm{tr}\Big(2T^{-1}(u)E_nT^{-1}(u)E_m\Big)\Big) + \mathrm{Re}\Big(\mathrm{tr}\Big(2T^{-1}(u)E_nT^{-1}(u)E_m^{\mathrm{H}}\Big)\Big)\,.$$

Define the $N \times N$ matrices A and B with entries

$$A_{n,m} = 2\,\mathrm{tr}\Big(T^{-1}(u)E_nT^{-1}(u)E_m\Big) \tag{C.59}$$

$$B_{n,m} = 2\,\mathrm{tr}\Big(T^{-1}(u)E_nT^{-1}(u)E_m^{\mathrm{T}}\Big). \tag{C.60}$$

Then the Hessian can be written in the form

$$\nabla_u^2 G(u) = \begin{pmatrix} \mathrm{Re}(A+B) & \mathrm{Re}(-jAJ^{\mathrm{T}}) \\ \mathrm{Re}(-jJA - jJB) & \mathrm{Re}(-JAJ^{\mathrm{T}} + JBJ^{\mathrm{T}}) \end{pmatrix}, \tag{C.61}$$

where J is a matrix which removes the first row, i.e., $J = (0, I)$, where 0 is a column of zeros and I is the $(N-1) \times (N-1)$ identity matrix.

At this point, we need a fast way of evaluating matrices A and B. Define the discrete Fourier transform matrix $W \in \mathbb{C}^{N_{\mathrm{FFT}} \times N}$ with entries

$$W_{n,m} = \exp(-j2\pi nm/N_{\mathrm{FFT}}), \tag{C.62}$$

where N_{FFT} is chosen such that $N_{\mathrm{FFT}} \geq 2N-1$. Recall that the Levinson-Durbin algorithm gives the decomposition $T^{-1}(u) = PDP^{\mathrm{H}}$, from which $T^{-1}(u) = RR^{\mathrm{H}}$ is obtained by calculating $R = PD^{\frac{1}{2}}$. Let S_n denote the discrete Fourier transform of the nth column of R (denote this column R_n), i.e., $S_n = WR_n$. Then by straight-forward generalization of the derivation in [34] to the complex-valued case, we get that A and B can be written in the forms

$$A = \frac{2}{N_{\mathrm{FFT}}^2} W^{\mathrm{T}} \left(\left(\sum_{l=0}^{N-1} S_l S_l^{\mathrm{H}} \right) \odot \left(\sum_{l=0}^{N-1} S_l S_l^{\mathrm{H}} \right) \right) W \tag{C.63}$$

$$B = \frac{2}{N_{\mathrm{FFT}}^2} W^{\mathrm{T}} \left(\left(\sum_{l=0}^{N-1} S_l S_l^{\mathrm{H}} \right) \odot \left(\sum_{l=0}^{N-1} S_l S_l^{\mathrm{H}} \right) \right) \bar{W}, \tag{C.64}$$

with $\odot$ denoting the Hadamard (entrywise) product. The forms (C.63)–(C.64) show that the Hessian of G can be evaluated in $\mathcal{O}(N^3)$ flops.

C.5.5 Evaluating the Diagonal Hessian Approximation

The L-BFGS approach uses the approximation of the Hessian diagonal (C.40) which requires calculation of the first entry of the Hessian

$$\Big(\nabla_u^2 h_t(u)\Big)_{0,0} = \Big(\nabla_u^2 g(u)\Big)_{0,0} + \frac{1}{t}\Big(\nabla_u^2 G(u)\Big)_{0,0}. \tag{C.65}$$

An $\mathcal{O}(N \log N)$ evaluation of the first term is easily obtained from (C.55). The second term can be evaluated based on (C.61), but a more efficient way can be found by proceeding as follows. From (C.58) we have

$$\Big(\nabla_u^2 G(u)\Big)_{0,0} = 4\,\mathrm{tr}\Big(T^{-1}(u)T^{-1}(u)\Big). \tag{C.66}$$

The matrix $T^{-1}(u)$ can be formed explicitly in $\mathcal{O}(N^2)$ flops using the Trench algorithm [48, 49]. However, since the decomposition (C.44) is already available in our setting, it is much easier to form $T^{-1}(u)$ as follows: Based on (C.44) we obtain for $n = 0, \ldots, N-1$ and $m = 0, \ldots, N-1-n$

$$\left(T^{-1}(u)\right)_{m,m+n} = \delta_{N-1}^{-1} \left(\sum_{k=0}^{m} \bar{\rho}_{N-1-k} \rho_{N-1-(k+n)} - \rho_{k-1} \bar{\rho}_{k+n-1} \right), \quad \text{(C.67)}$$

i.e., $T^{-1}(u)$ is "formed along the diagonals". By implementing the above sum as a cumulative sum, the complete matrix $T^{-1}(u)$ is formed in $\mathcal{O}(N^2)$ flops. Note that since $T(u)$ is both Hermitian and persymmetric, then so is $T^{-1}(u)$. This means that only one "wedge" of the matrix, about $N/4$ entries, must be calculated explicitly [49].

The trace in (C.66) is evaluated by taking the magnitude square of all entries in $T^{-1}(u)$ and summing them.

C.5.6 Analysis of Computational Complexity

To summarize the computational complexity of Algorithm C.1 based on the low-complexity evaluations above, consider the two methods for determining the search direction.

- Using Newtons method: The computation time is asymptotically dominated by the evaluation and inversion of the Hessian, which takes $\mathcal{O}(N^3)$ flops.
- Using L-BFGS: The computation time is asymptotically dominated by the $\mathcal{O}(MN)$ modified L-BFGS two-loop recursion in Alg. C.2 or by the $\mathcal{O}(N^2)$ evaluation of the diagonal Hessian approximation.

When using the Newton search direction, the decomposition (C.47) is required and the Levinson-Durbin algorithm must therefore be used to evaluate the factorization of the Toeplitz inverse. When using the L-BFGS search direction either the generalized Schur or the Levinson-Durbin algorithm can be used. The choice does not affect the asymptotic computational complexity, but one may be faster than the other in practice.

C.6 Numerical Experiments

C.6.1 Setup & Algorithms

In our experiments we use the signal model (C.2). The frequencies $\omega_0, \ldots, \omega_{K-1}$ are drawn randomly on $[0, 2\pi)$, such that the minimum separation[6] between

[6] The wrap-around distance on $[0, 2\pi)$ is used for all differences of frequencies.

Variant	L-BFGS	Newton
Number of saved difference vectors M	$2N-1$	-
Armijo parameter c	0.05	0.05
Barrier parameter multiplier γ	2	10
Absolute tolerance ε_{abs}	10^{-4}	10^{-7}
Relative tolerance ε_{rel}	10^{-4}	10^{-7}

Table C.1: Algorithm parameters.

any two frequencies is $4\pi/N$. The coefficients $c_0, \ldots, c_{K-1}$ are generated i.i.d. random according to a circularly symmetric standard complex Gaussian distribution. After generating the set of K frequencies and coefficients, the variance of the noise vector ζ is selected such that the desired signal-to-noise ratio (SNR) is obtained. The regularization parameter τ is selected from (C.9) based on the true noise variance. We assess the algorithms based on their ability to solve AST, which is obtained by selecting $w = 2e_0$ in (C.1).

We dub our proposed algorithm as FastAST [7] and show results using both L-BFGS and Newtons method to calculate the search direction. For $N \leq 512$ our implementation uses the Levinson-Durbin algorithm for Toeplitz inversion, while for $N > 512$ it uses the generalized Schur algorithm where applicable. The parameters of the algorithm are listed in Table C.1. It is worth to say a few words about the number of saved difference vectors M in L-BFGS. On the one hand, selecting larger values of M can also decrease the total number of iterations required by improving the Hessian approximation, but on the other hand doing so increases the number of flops required per iteration. In our numerical experiments we have found that setting it equal to the size of u ($M = 2N - 1$) provides a good trade-off. Loosely speaking this choice allows L-BFGS to perform a full-rank update of the Hessian approximation, while it does not increase the asymptotic per-iteration computational complexity. With this choice the algorithm asymptotically requires $\mathcal{O}(N^2)$ flops per iteration.

Performance of the ADMM algorithm[8] [9] is also shown along with that of CVX [50] using the SeDuMi backend [18].

C.6.2 Solution Accuracy Per Iteration

For this investigation a ground-truth solution of (C.1) is obtained using CVX with the precision setting set to "best". We denote this value as $\mu^\star$. Fig. C.1 shows the normalize squared error between $\mu^\star$ and the solution in each iteration of the algorithms. The algorithms ignore the stopping criteria and run

[7] We have published our implementation at github.com/thomaslundgaard/fast-ast.
[8] We use the implementation from github.com/badrinarayan/astlinespec.

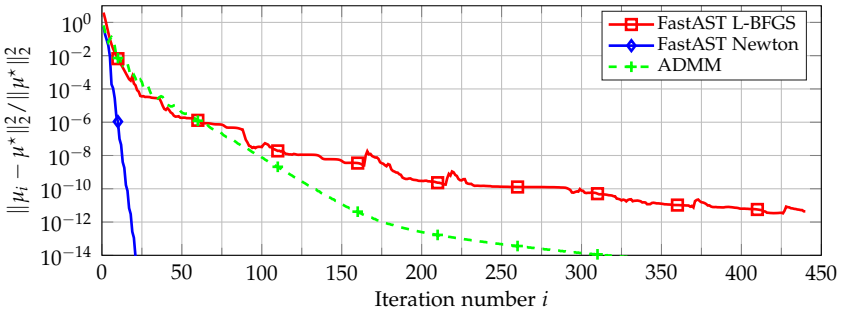

Fig. C.1: Solution accuracy per iteration. The signal length is $N = 64$, the number of sinusoids is $K = 6$ and the SNR is 20 dB.

until no further progress can be made towards the solution.

FastAST Newton converges very fast and a solution of very high accuracy is obtained within 25 iterations. This is due to the well-known quadratic convergence of Newtons method. While FastAST L-BFGS converges significantly slower it requires only $\mathcal{O}(N^2)$ flops per iteration versus the $\mathcal{O}(N^3)$ flops per iteration of FastAST Newton. We therefore cannot, at this point, conclude which version of FastAST is faster in practice. Note that ADMM on the other hand requires $\mathcal{O}(N^3)$ flops per iteration, the same as FastAST Newton, but requires significantly more iterations.

It is seen that FastAST L-BFGS cannot progress more after iteration 440. This happens due to numerical challenges in evaluating the L-BFGS search direction. It is well-known that Woodbury's matrix identity, upon which L-BFGS is based, has limited numerical stability. For this reason FastAST L-BFGS is unable to obtain a solution of the same accuracy as CVX. Despite of this, as seen in the following sections, the solution accuracy of FastAST L-BFGS is sufficiently high in all cases but those with very high SNR. The tolerance values of FastAST L-BFGS are selected larger than for FastAST Newton (Table C.1) because of the mentioned numerical issues with obtaining a high-accuracy solution.

FastAST Newton does not suffer from this problem and can obtain a solution of about the same accuracy as CVX. ADMM can also obtain a solution of high accuracy but, as can be seen in Fig. C.1, it has slow convergence starting around iteration number 175. It therefore takes a large number of iterations to obtain a solution of the same accuracy as CVX or FastAST Newton.

C.6.3 Metrics

In the following we perform a Monte Carlo simulation study. Four metrics of algorithm performance and behaviour are considered: normalized mean-square error (NMSE) of the reconstructed signal x; mean-square error (MSE)

of the frequencies conditioned on successful recovery; number of iterations and algorithm runtime. The NMSE of the reconstructed signal is obtained by estimating the frequencies from the dual polynomial as described in [9] and using these to obtain the least-squares solution for the coefficients. An estimate of x is then obtained by inserting into (C.3). This estimate is also known as the *debiased* solution and it is known to have smaller NMSE than the estimate of x directly obtained as the solution of (C.1) [9]. In the evaluation of the signal reconstruction the performance of an Oracle estimator is also shown. The Oracle estimator knows the true frequencies and estimates the coefficients using least-squares.

To directly assert the accuracy with which the frequencies are estimated we present the MSE of the frequency estimates obtained from the dual polynomial. The MSE of the frequency estimates is only calculated based on these Monto Carlo trails in which the set of frequencies has been successfully recovered. Successful recovery is understood as correct estimation of the model order K and that all frequency estimates are within a distance of π/N from their true value. The association of the estimated to the true frequencies is obtained by minimizing the frequency MSE using the Hungarian method [51] (also known as Munkres assignment algorithm).

The simulations are performed on a 2011 MacBook Pro using MATLAB R2016b. MATLAB is restricted to only use a single CPU core, such that the runtime of the algorithms can be compared without differences in the parallelism achieved in the implementations. The computationally heavy steps of both FastAST and ADMM are implemented in native code using the automatic code generation ("codegen") feature of MATLAB.

C.6.4 Performance Versus Problem Size

The performance versus problem size N is shown in Fig. C.2. We first note that all algorithms give the same estimation accuracy at all problem sizes, providing strong evidence that they correctly solve (C.1). The number of iterations of FastAST L-BFGS increases with N. It is then expected that the total runtime asymptotically scales at a rate above the per-iteration cost of $\mathcal{O}(N^2)$ flops. Even still, the runtime for N up to $2,048$ scales at a rate of about $\mathcal{O}(N^2)$.

The number of iterations of FastAST Newton is practically independent of N. We therefore expect the total runtime to scale asymptotically as $\mathcal{O}(N^3)$. In practice it scales a little better for the values of N considered here. The number of iterations of ADMM increases significantly with N (doubling N roughly doubles the number of iterations). This in turn means that the runtime scales faster than the asymptotic per-iteration cost of $\mathcal{O}(N^3)$ flops.

In conclusion both variants of FastAST are faster than ADMM already at $N = 128$ and their runtime scales at a rate much slower than ADMM. This

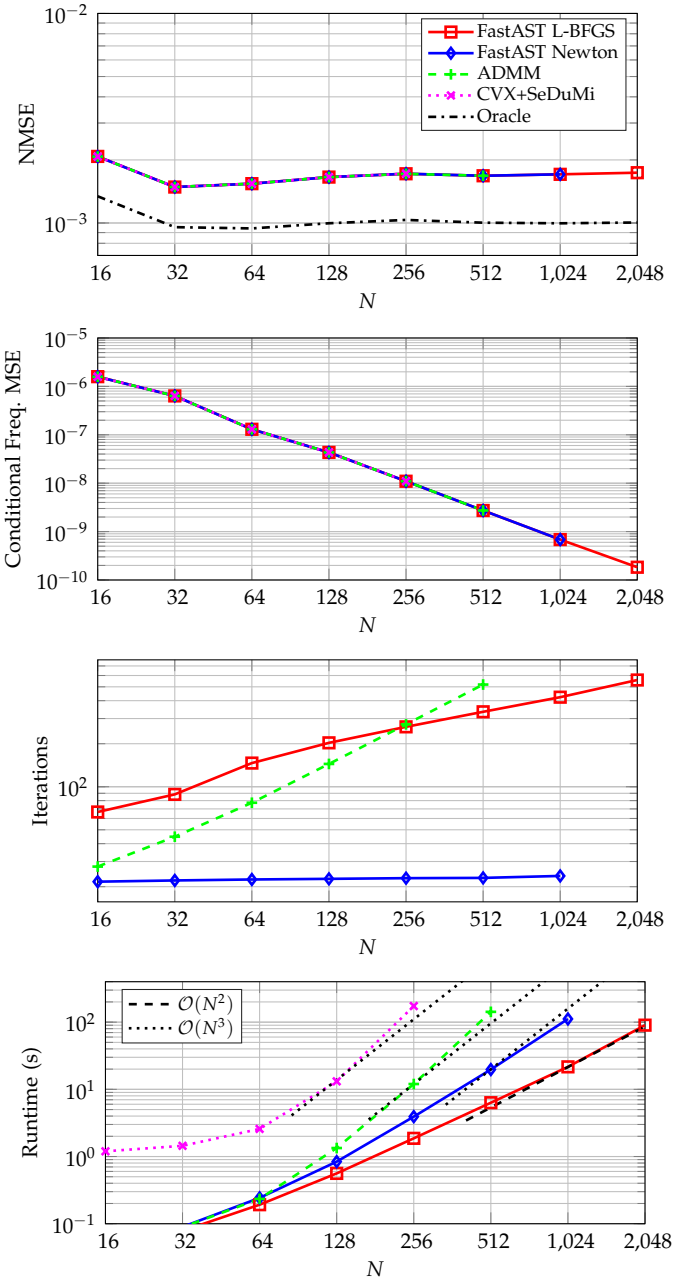

Fig. C.2: Simulation results for varying problem size N. The SNR is 20 dB and the number of sinusoids K is selected as $N/10$ rounded to the nearest integer. Results are averaged over 100 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown. In the figure with runtime the asymptotic per-iteration computational complexity is also plotted.

means that they are significantly faster than ADMM for large values of N. In large N it is also clear that the L-BFGS variant of FastAST is significantly faster than the Newton variant.

C.6.5 Performance Versus Signal-to-Noise Ratio

Fig. C.3 shows performance versus the SNR level. Note that the conditional MSE of the frequency estimates is not shown for 0 dB SNR because there are no Monte Carlo trials with successful recovery of the frequencies at this SNR.

At SNR up to 30 dB all the algorithms perform the same in terms of NMSE of x and conditional MSE of the frequency estimates. This means that all algorithms have found a sufficiently accurate solution of (C.1) (relative to the SNR). In SNR larger than 30 dB FastAST L-BFGS shows a degraded solution accuracy compared to the remaining algorithms. This is because of the mentioned numerical issues and the consequently larger tolerances selected (cf. Table C.1).

In terms of number of iterations and runtime note that both variants of FastAST show roughly unchanged behaviour with different SNR. ADMM on the other hand requires much larger number of iterations and larger runtime for large SNR. In large SNR it is evident that FastAST Newton is preferred due to lower runtime than ADMM and higher estimation accuracy than FastAST L-BFGS.

C.7 Conclusions

The FastAST algorithm presented in this paper provides a fast approach to solving the atomic norm soft thresholding problem (C.1). The L-BFGS variant provides a reasonably accurate solution and is much faster than any other algorithm for large problem size N. If a solution of high accuracy is requested, which may be desirable in very high SNR, a variant of FastAST based on Newtons method is also provided. This variant can find a solution of high accuracy in a small number of iterations. While it is slower than FastAST L-BFGS, it is significantly faster than the state-of-the-art method based on ADMM.

The FastAST algorithm is obtained by reformulating the semidefinite program (C.1) as a non-symmetric conic program (C.12). This reformulation is of key importance in obtaining a fast algorithm. This work has provided an example of an optimization problem where it is beneficial to formulate it as a non-symmetric conic program instead of the standard, and much better understood, formulation as a symmetric conic program. We have also provided an implementation of a non-symmetric conic solver, thereby demonstrating the practical feasibility of this class of methods.

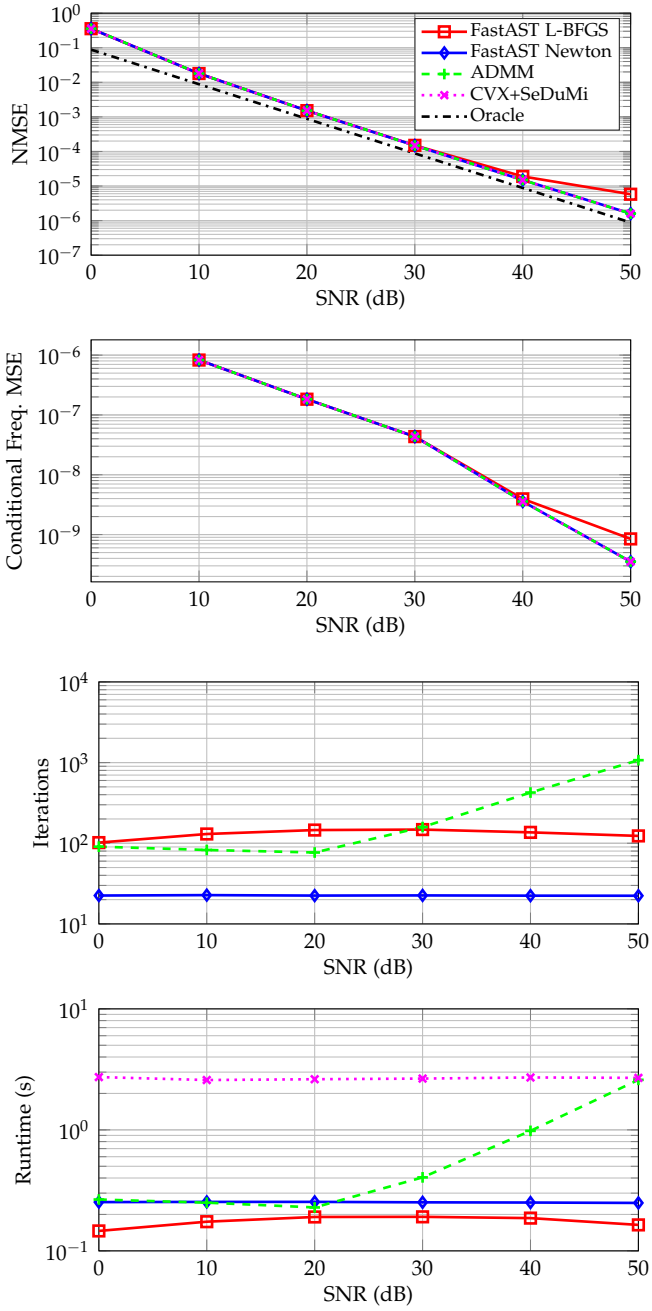

Fig. C.3: Simulation results for varying SNR. The signal length is $N = 64$ and the number of sinusoids is $K = 6$. Results are averaged over 100 Monte Carlo trials. The legend applies to all plots. Only the NMSE of Oracle is shown.

In one of the variants of FastAST the L-BFGS method is used for the solution of the barrier problem (C.35). We have demonstrated that L-BFGS can be used to obtain a quasi-Newton method for the barrier method despite the fact that the barrier parameter is updated in every iteration. This approach can directly be applied in other solvers based on the barrier method.

Finally note that there are many examples of optimization problems of practical interest which involve a constraint in either the cone of finite autocorrelation sequences $\mathcal{C}^*$ or the cone $\mathcal{K}$. An example is the gridless SPICE method [15] for line spectral estimation; or frequency-domain system identification and filter design as summarized in [34]. We expect that equally fast primal-dual interior point methods can be derived for all of these problems using the techniques of this paper. We also expect that it is fairly straightforward to extend FastAST to atomic norm minimization with partial observations [8] or multiple measurement vectors [14]. An interesting, but less obvious, extension is to the multi-dimensional harmonic retrieval problem [52]; for that purpose the work [53] may contain some useful insights.

Acknowledgements

We would like to thank Lieven Vandenberghe and Martin Skovgaard Andersen for providing valuable input to the work and pointing us to some important references.

The work of T. L. Hansen is supported by the Danish Council for Independent Research under grant id DFF–4005–00549.

A Characterization of $\mathcal{K}^*$

To characterize the dual cone $\mathcal{K}^*$, a number of lemmas are needed.

Lemma C.3. *Let K by a proper cone and assume $\lambda \neq 0$. If $\langle \lambda, \mu \rangle \geq 0$ for every $\mu \in \mathbf{int}\,\mathcal{K}$, then $\langle \lambda, \mu \rangle \geq 0$ for every $\mu \in \mathcal{K}$.*

Proof. Let $\tilde{\mu} \in \mathcal{K}$ and let $\{\mu_i\}$ be a sequence which converges to $\tilde{\mu}$ with $\mu_i \in \mathbf{int}\,\mathcal{K}$. Then $\langle \lambda, \mu_i \rangle \geq 0$ and so $\langle \lambda, \tilde{\mu} \rangle = \lim_{i \to \infty} \langle \lambda, \mu_i \rangle \geq 0$, completing the proof. □

Lemma C.4. *Let $\mathcal{K}$ be a proper cone. The interior of its dual is given by*

$$\mathbf{int}\,\mathcal{K}^* = \{\lambda : \langle \lambda, \mu \rangle > 0 \ \ \forall\, \mu \in \mathcal{K}\}. \tag{C.68}$$

Proof. See [17], exercise 2.31. □

To formulate the next lemma, the dual barrier of F is introduced:

$$F^*(\lambda) = \sup\{-\langle\lambda,\mu\rangle - F(\mu) : \mu \in \mathbf{int}\,\mathcal{K}\}. \tag{C.69}$$

This function is a slight modification ($-\langle\lambda,\mu\rangle$ replaces $\langle\lambda,\mu\rangle$) of the convex conjugate of F. It turns out that F^* is a LH barrier function for the dual cone $\mathcal{K}^*$ [17, 20]. Its usefulness for our purposes lies in the following property.

Lemma C.5. *Assume $\lambda \neq 0$ and let $\mathcal{K}$ be a proper cone with corresponding LH barrier function F. Then $\lambda \in \mathbf{int}\,\mathcal{K}^*$ if and only if $F^*(\lambda) < \infty$ (i.e., F^* is bounded above.)*

Proof. We first prove the direct implication. Reasoning by contradiction, assume that $F^*(\lambda) < \infty$ and that there exists a $\mu \in \mathbf{int}\,\mathcal{K}$ such that $\langle\lambda,\mu\rangle < 0$. Then $\alpha\mu \in \mathbf{int}\,\mathcal{K}$ for all $\alpha > 0$. But $\lim_{\alpha\to\infty} -\langle\lambda,\alpha\mu\rangle - F(\alpha\mu) = \lim_{\alpha\to\infty} -\alpha\langle\lambda,\mu\rangle - F(\mu) + \theta_F\log(\alpha) = \infty$, a contradiction, so $\langle\lambda,\mu\rangle \geq 0$ for every $\mu \in \mathbf{int}\,\mathcal{K}$. By Lemma C.3 we have $\langle\lambda,\mu\rangle \geq 0$ for every $\mu \in \mathcal{K}$, thus $\lambda \in \mathcal{K}^*$. Since F^* is a LH barrier function for $\mathcal{K}^*$, it is easy to show that $F^*(\lambda) < \infty$ implies $\lambda \notin \mathbf{bd}\,\mathcal{K}^*$, so $\lambda \in \mathbf{int}\,\mathcal{K}^*$.

To prove the converse assume $\lambda \in \mathbf{int}\,\mathcal{K}^*$. Then by Lemma C.4, we have $\langle\lambda,\mu\rangle > 0$ for all $\mu \in \mathcal{K}$. It follows that there exists an $\varepsilon > 0$ such that $\langle\lambda,\tilde{\mu}\rangle \geq \varepsilon$ for every $\tilde{\mu} \in \mathcal{K}$ with $\|\tilde{\mu}\|_2 = 1$. By continuity of F it can also be shown that there exists a δ such that $F(\tilde{\mu}) \geq \delta$ for every $\tilde{\mu} \in \mathcal{K}$ with $\|\tilde{\mu}\|_2 = 1$. With $\tilde{\mu} = \mu/\|\mu\|_2$, the objective in (C.69) obeys

$$\begin{aligned} -\langle\lambda,\mu\rangle - F(\mu) &= -\|\mu\|_2\langle\lambda,\tilde{\mu}\rangle - F(\|\mu\|_2\tilde{\mu}) && \text{(C.70)}\\ &= -\|\mu\|_2\langle\lambda,\tilde{\mu}\rangle - F(\tilde{\mu}) + \theta_F\log(\|\mu\|_2) && \text{(C.71)}\\ &\leq -\|\mu\|_2\varepsilon - \delta + \theta_F\log(\|\mu\|_2). && \text{(C.72)} \end{aligned}$$

The second equality follows from logarithmic homogeneity of F. This function is bounded above and so $F^*(\lambda) < \infty$. □

We are now ready to give the desired proof.

Proof of Lemma C.1. It is easy to show the following:

1. If $\rho < 0$, then $\lambda \notin \mathcal{K}^*$.
2. If $\rho = 0$ and $s \neq 0$, then $\lambda \notin \mathcal{K}^*$.
3. If $\rho = 0$ and $s = 0$, then $\lambda \in \mathcal{K}^*$ if and only if $z \in \mathcal{C}^*$.

The first and second property are shown by constructing a $\mu \in \mathcal{K}$ such that $\langle\lambda,\mu\rangle < 0$. The third property is shown by writing $\langle\lambda,\mu\rangle = z^{\mathrm{T}}u \geq 0$ for all $\mu \in \mathcal{K}$ if and only if $z^{\mathrm{T}}u \geq 0$ for all $u \in \mathcal{C}$.

The only case we have not considered so far is $\rho > 0$. By Lemma C.5 and (C.24), we have $\lambda \in \mathbf{int}\,\mathcal{K}^*$ if and only if $F^*(\lambda) < \infty$, i.e., when

$$h(\mu) = -\rho v - \mathrm{Re}(s^{\mathrm{H}}x) - z^{\mathrm{T}}u + \log|T(u)| + \log(v - x^{\mathrm{H}}T^{-1}(u)x) \qquad \text{(C.73)}$$

is bounded above on the domain $\mu \in \mathbf{int}\,\mathcal{K}$. The function h is concave and by setting the gradient equal to zero we get optimal points

$$\begin{aligned} v^\star &= \rho^{-1} + (2\rho)^{-2} s^{\mathrm{H}} T(u) s \\ x^\star &= -(2\rho)^{-1} T(u) s. \end{aligned}$$

It is easy to show that if $u \in \mathbf{int}\,\mathcal{C}$, then $(v^\star, x^\star, u)^{\mathrm{T}} \in \mathbf{int}\,\mathcal{K}$. Inserting into $h(\mu)$ we obtain

$$\begin{aligned} h(\mu) &\leq -z^{\mathrm{T}}u + \frac{1}{4\rho} s^{\mathrm{H}} T(u) s - 1 - \log(\rho) + \log|T(u)| \\ &= -c^{\mathrm{T}}u - 1 - \log(\rho) + \log|T(u)|, \end{aligned}$$

with $c(\lambda) = z - \frac{1}{4\rho} T^*(ss^{\mathrm{H}})$. For each $u \in \mathbf{int}\,\mathcal{C}$ there exists some corresponding $\mu \in \mathbf{int}\,\mathcal{K}$ such that the above holds with equality.

If $c(\lambda) = 0$, the function $h(\mu)$ is unbounded above on the domain $\mu \in \mathbf{int}\,\mathcal{K}$ and so $\lambda \notin \mathbf{int}\,\mathcal{K}^*$. If $c(\lambda) \neq 0$ we can use Lemma C.5 because $-\log|T(u)|$ is a LH barrier function for $\mathcal{C}$. So $h(\mu)$ is bounded above on the domain $\mu \in \mathbf{int}\,\mathcal{K}$ if and only if $c(\lambda) \in \mathbf{int}\,\mathcal{C}^*$. Tracing back our steps above we have (for $\rho > 0$) that $\lambda \in \mathbf{int}\,\mathcal{K}^*$ if and only if $c(\lambda) \in \mathbf{int}\,\mathcal{C}^*$. Since both of the dual cones are closed sets and $c(\cdot)$ is a continuous function, we have $\lambda \in \mathcal{K}^*$ if and only if $c \in \mathcal{C}^*$. That completes the proof. □

References

[1] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Found. Comp. Math.*, vol. 12, no. 6, pp. 805–849, 2012.

[2] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Methods in Convex Programming*. SIAM, 1994.

[3] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[4] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 1999.

[5] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[6] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[7] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21 – 30, Mar. 2008.

[8] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7465–7490, Nov 2013.

[9] B. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with application to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, Dec 2013.

[10] G. Tang, B. N. Bhaskar, and B. Recht, "Near minimax line spectral estimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 499–512, Jan 2015.

[11] E. J. Candés and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Commun. Pure Appl. Math.*, vol. 67, no. 6, pp. 906–956, 2014.

[12] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *J. Fourier Anal. Applicat.*, vol. 19, no. 6, pp. 1229–1254, 2013.

[13] M. Cho, K. V. Mishra, J. F. Cai, and W. Xu, "Block iterative reweighted algorithms for super-resolution of spectrally sparse signals," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2319–2313, Dec 2015.

[14] Y. Li and Y. Chi, "Off-the-grid line spectrum denoising and estimation with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1257–1269, Mar. 2016.

[15] Z. Yang and L. Xie, "On gridless sparse methods for line spectral estimation from complete and incomplete data," *IEEE Trans. Signal Process.*, vol. 63, no. 12, pp. 3139–3153, June 2015.

[16] M. F. Da Costa and W. Dai, "Low dimensional atomic norm representations in line spectral estimation," in *IEEE Int. Symp. Inform. Theory*, 2017, pp. 226–230.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[18] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11-12, pp. 625–653, 1999.

[19] K. Toh, M. Todd, and R. Tutuncu, "SDPT3 — a Matlab software package for semidefinite programming," *Opt. Methods Softw.*, vol. 11, pp. 545–581, 1999.

[20] Y. E. Nesterov and M. J. Todd, "Self-scaled barriers and interior-point methods for convex programming," *Math. Oper. Res.*, vol. 22, no. 1, pp. 1–42, 1997.

[21] ——, "Primal-dual interior-point methods for self-scaled cones," *SIAM J. Opt.*, vol. 8, no. 2, pp. 324–364, 1998.

[22] O. Güler, "Barrier functions in interior point methods," *Math. Oper. Res.*, vol. 21, no. 4, pp. 860–885, 1996.

[23] Y. Nesterov, "Towards non-symmetric conic optimization," *Opt. Methods Softw.*, vol. 27, no. 4-5, pp. 893–917, 2012.

[24] A. Skajaa, J. B. Jørgensen, and P. C. Hansen, "On implementing a homogeneous interior-point algorithm for nonsymmetric conic optimazation," Technical University of Denmark, Tech. Rep. IMM–2011-02, 2011.

[25] A. Skajaa and Y. Ye, "A homogeneous interior-point algorithm for nonsymmetric convex conic optimization," *Math. Prog.*, vol. 150, no. 2, pp. 391–422, 2015.

[26] L. Tunçel, "Generalization of primal-dual interior-point methods to convex optimization problems in conic form," *Found. Comp. Math.*, vol. 1, no. 3, pp. 229–254, 2001.

[27] G. S. Ammar and W. B. Gragg, "The generalized Schur algorithm for the superfast solution of Toeplitz systems," in *Rational Approx. Applicat. Math. Phys.* Springer, 1987, pp. 315–330.

[28] ——, "Numerical experience with a superfast real Toeplitz solver," *Linear Algebra Applicat.*, vol. 121, pp. 185–206, Aug. 1989.

[29] N. Levinson, "The Wiener (root mean square) error criterion in filter design and prediction," *Stud. Appl. Math.*, vol. 25, no. 1-4, pp. 261–278, 1946.

[30] J. Durbin, "The fitting of time-series models," *Revue de l'Institut Int. de Statistique*, pp. 233–244, 1960.

[31] T. L. Hansen, B. H. Fleury, and B. D. Rao, "Superfast line spectral estimation," 2018, in press, IEEE Trans. Signal Process. http://arxiv.org/abs/1705.06073.

[32] B. R. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 5, pp. 1333–1335, Oct. 1985.

[33] Y. Genin, Y. Hachez, Y. Nesterov, and P. Van Dooren, "Optimization problems over positive pseudopolynomial matrices," *SIAM J. Matrix Anal. Applicat.*, vol. 25, no. 1, pp. 57–79, 2003.

[34] B. Alkire and L. Vandenberghe, "Convex optimization problems involving finite autocorrelation sequences," *Math. Prog.*, vol. 93, no. 3, pp. 331–359, Dec. 2002.

[35] A. Nemirovski, "Lecture notes: Interior point polynomial time methods in convex programming," 2004, Georgia Institute of Technology.

[36] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug 2005.

[37] B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Process.*, vol. 40, pp. 590–600, Mar. 1992.

[38] R. Carriere and R. L. Moses, "High resolution radar target modeling using a modified Prony estimator," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 13–18, Jan. 1992.

[39] W. Bajwa, A. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, Jun. 2010.

[40] X. Andrade, J. N. Sanders, and A. Aspuru-Guzik, "Application of compressed sensing to the simulation of atomic systems," *Proc. Nat. Academy Sciences*, vol. 109, no. 35, pp. 13 928–13 933, Jul. 2012.

[41] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Stat. Soc., Ser. B*, vol. 58, pp. 267–288, 1994.

[42] Z. Yang and L. Xie, "Enhancing sparsity and resolution via reweighted atomic norm minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 995–1006, Feb 2016.

[43] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Applicat.*, vol. 14, no. 5-6, pp. 877–905, 2008.

[44] M. G. Krein and A. A. Nudelman, *The Markov moment problem and extremal problems*, ser. Translations of Mathematical Monographs. American Mat. Soc., 1977, vol. 50.

[45] A. Nemirovski and M. J. Todd, "Interior-point methods for optimization," *Acta Numerica*, vol. 17, pp. 191–234, may 2008.

[46] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Math. Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[47] I. Gohberg and I. A. Feldman, *Convolution equations and projection methods for their solution*, ser. Translations of Mathematical Monographs. American Mat. Soc., 2005, vol. 41.

[48] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices," *J. Soc. Ind. Appl. Math.*, vol. 12, no. 3, pp. 515–522, 1964.

[49] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[50] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," http://cvxr.com/cvx/, Jan. 2018.

[51] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1-2, pp. 83–97, 1955.

[52] Y. Chi and Y. Chen, "Compressive two-dimensional harmonic retrieval via atomic norm minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1030–1042, Feb. 2015.

[53] Z. Yang, L. Xie, and P. Stoica, "Vandermonde decomposition of multilevel Toeplitz matrices with application to multidimensional super-resolution," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3685–3701, June 2016.

SUMMARY

Line spectral estimation is a classical problem in signal processing. It has found broad application in for example array processing, wireless communication, localization, radar, radio astronomy and audio. In the last decade we have seen significant research into sparsity-based processing techniques. The use of sparsity-based techniques has allowed for advances to both the design and analysis of algorithms for line spectral estimation. In this thesis we study the design of such algorithms.

The uniting theme of our contributions is the design of algorithms that make sparsity-based line spectral estimation viable in practice. First it is demonstrated that these schemes can be applied to the estimation of wireless channels of not only specular but also of diffuse nature. We attribute that to a low-rank property of the channel covariance matrix, a concept that we elaborate on.

The design of algorithms for sparsity-based line spectral estimation in a general context is then considered. The obtained algorithms are computationally feasible for much larger problems than what concurrent algorithms can practically deal with and show high estimation accuracy.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-170-5

AALBORG **UNIVERSITY PRESS**